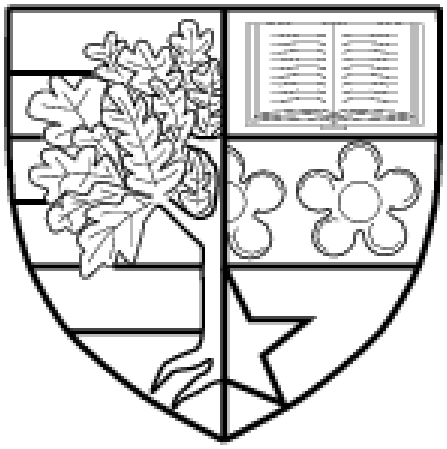
Investigating Features and Techniques for Arabic Authorship Attribution

By

Kareem Shaker



Submitted for the degree of

Doctor Of Philosophy

On compilation of research in the

Department Of Computer Science

School of Mathematics and Computer Science

Heriot-Watt University

March 2012

Abstract

Authorship attribution is the problem of identifying the true author of a disputed text. Throughout history, there have been many examples of this problem concerned with revealing genuine authors of works of literature that were published anonymously, and in some cases where more than one author claimed authorship of the disputed text. There has been considerable research effort into trying to solve this problem. Initially these efforts were based on statistical patterns, and more recently they have centred on a range of techniques from artificial intelligence. An important early breakthrough was achieved by Mosteller and Wallace in 1964 [15], who pioneered the use of 'function words' – typically pronouns, conjunctions and prepositions – as the features on which to base the discovery of patterns of usage relevant to specific authors.

The authorship attribution problem has been tackled in many languages, but predominantly in the English language. In this thesis the problem is addressed for the first time in the Arabic Language. We therefore investigate whether the concept of functions words in English can also be used in the same way for authorship attribution in Arabic. We also describe and evaluate a hybrid of evolutionary algorithms and linear discriminant analysis as an approach to learn a model that classifies the author of a text, based on features derived from Arabic function words. The main target of the hybrid algorithm is to find a subset of features that can robustly and accurately classify disputed texts in unseen data. The hybrid algorithm also aims to do this with relatively small subsets of features. A specialised dataset was produced for this work, based on a collection of 14 Arabic books of different natures, representing a collection of six authors. This dataset was processed into training and test partitions in a way that provides a diverse collection of challenges for any authorship attribution approach.

The combination of the successful list of Arabic function words and the hybrid algorithm for classification led to satisfying levels of accuracy in determining the author of portions of the texts in test data. The work described here is the first (to our knowledge) that investigates authorship attribution in the Arabic knowledge using computational methods. Among its contributions are: the first set of Arabic function words, the first specialised dataset aimed at testing Arabic authorship attribution methods, a new hybrid algorithm for classifying authors based on patterns derived from these function words, and, finally, a number of ideas and variants regarding how to use function words in association with character level features, leading in some cases to more accurate results.

Acknowledgment

First, I thank God for all what I accomplished.

I would like to thank my parents for their great support in every way and manner. They were with me in every step supporting and encouraging me with all their love, means and efforts. They visited me in UK many times and they phoned me every single night to make sure that me and my small family are ok. I am really grateful and thankful for them. I hope one day I can reward them in return

I also thank my wife Marwa. I can't find words to express how helpful, supporting, encouraging and caring she was in all the time during my studies. She was always there in the tough and hard times and never pulled me back in any way. Thank you again for keeping our small family together happy and well looked after.

I want to thank my kids Lojan and Yassin for their patient and expressing their love to me all the time. They also showed me their understanding for all the busy time I was away working and not with them.

I also thank my sister Nermin and brothers Hassan, Amr and Hossam for supporting me by showing great care and love.

I am thankful too to my wife's family for love and care. They were always there when needed.

Finally, I would like to thank my Supervisor Prof David W. Corne for his help and great guidance throughout the research. His knowledge and experience added a lot and taught me great stuff that helped me complete the work and come out in this great form.

Without you all I couldn't accomplish all this.

Contents

List of Tables

List of Figures

.

1

General Introduction

Chapter overview

In this chapter we introduce the Authorship Attribution problem. In section 1.1 we provide a general definition of the Authorship Attribution problem. In section 1.2 we introduce the main three types of Authorship Attribution Problems. In section 1.3 we present the obstacles and difficulties faced in solving the problem of Authorship Attribution. In section 1.4 we introduce the reasons behind publishing anonymous texts and books (i.e. these are some of the situations that lead to an Authorship Attribution problem). In section 1.5 we describe text features, and how they reflect the author’s 'fingerprint'. In section 1.6 we describe the organisation of the remainder of the thesis and we describe the contributions of this thesis.

1.1 What Is An Authorship Attribution Problem?

Authorship Attribution is the problem of identifying the genuine author of a disputed work. This could be the basic definition of Authorship Attribution. But there are many types and forms of this problem. The types vary according to the circumstances and incidents that have occurred to create ambiguity in regard to the authorship of one piece of work. No one can be clear or sure about the true author of a disputed work, except by introducing evidence. There are two types of evidence according to Don Foster in the introduction of his book, Author Unknown [11]. One is known as external evidence, which are the facts collected and built around the unattributed text, such as the date of publication, place of publication, drafts or any piece of information related to the text. The other type of evidence is known as internal evidence. Examples of internal evidence are the information that could be found and inferred from the heart of the text itself, such as the style and manners of the true author which could be found in his or her usage of words, vocabulary, punctuation and the organization of the text. Scholars and researchers tend to look for the internal evidence as a tool to reveal the genuine author, in cases where the external evidence is not enough or not satisfactory. There are many cases of Authorship Attribution problem where the body of internal evidence is considered the main and the only way to solve the problem. The level of difficulty depends on the type of the Authorship Attribution problem. There are three main types of Authorship Attribution.

1.2 Types Of Authorship Attribution Problem

The basic form and type of Authorship Attribution problem is when two writers claim the authorship of one disputed work. It is considered the simplest because the argument is between only two known writers or authors. Therefore, the investigation is focused on these two authors and the comparison between their way of writing or style and the style of the disputed work will be straightforward. An example of this type of Authorship Attribution case appeared over the controversy of the disputed poem *A visit from St. Nicolas* or *Twas the night before Christmas.* Another famous case of disputed work in English literature that falls into the simplest type of Authorship Attribution is the 15th book of OZ.

Another type of authorship attribution problem appears when some investigators and researchers examine an anonymous work and try to find the actual writer of it. This type of the problem is considered much more complicated than the basic one which only compares between two known authors. The investigation usually starts with a pool or a group of suspected or supposed authors. Then, they usually end up with two or more authors that researchers and investigators believe these authors were contemporary with and existing in the same time the anonymous text appeared and the anonymous text matches their way and style of writing.

One of the typical cases of this type of Authorship Attribution problem is the *Primary Colours* novel which was published anonymously. This case is an example of how the Authorship Attribution problem could be significantly more complicated than the 'two-writer' case. And also the case of *The Funeral Elegy by W.S.* which is a piece of work that was only signed "W.S."

Moreover, another situation that introduces the Authorship Attribution problem happens when researchers suspect and doubt the originality of some texts, according to their belief that such a writer or author can’t be the genuine author because of some evidence that they provide. An example is the letters of G. Pickett Letters in the A *widow and her soldier.*

1.3 Obstacles to The Discovery Of The True Author

There are obstacles and problems that make the process of solving the Authorship Attribution problem complicated. The main problem, or the most obvious one, is the alteration or the editing of the original text. The problem could be easy, or the real author could be determined if the original words of the disputed work are not altered or changed in any way. If the original text is untouched, or only lightly edited, it should contain all the features that can be used to solve the problem, such as the true author's genuine vocabulary, punctuation, spelling, sentence structure and grammar usage. All these features could be found in the disputed work and can be used to extract the fingerprint of the authors. Then, these fingerprints could be compared to other texts that lead to the true author. But if the original text is changed or transformed, this may lead to distorted fingerprints and inaccurate results. There are many famous cases of authorship attribution in the history of literature in which the disputed works were not altered by any of the claiming authors. Most of these works were originally published anonymously, so there was no need for any alteration of the original texts such as in *The Federalist papers*, the poem *a visit from St. Nicolas* and the Novel *Primary Colours*. While other cases of the problem appeared after the publication, such as in the *George Pickett Letters*, *The 15th book of Oz* and *Funeral Elegy by W.S.* Therefore, the investigation in such cases is straightforward, being only the problem of identifying who is the real writer.

On the other hand, nowadays, cases are much complicated when the "idea" of a book, scientific paper or a literature work is stolen, and the material is altered in some way or another, so that the real author cannot maintain his possession by proving that the words used belongs to him. Stealing an idea and building a text based on this idea is one of the common problems facing Authorship Attribution because it is very hard to prove. This problem has a great impact on the educational system. Many researches and experimental ideas have been stolen, and the texts edited and otherwise changed, so that the original researcher could not prove his authorship of his work. In this type of case, the only way to find the truth is via external evidence, which may come from people who worked on the same project, or any other evidence that pertains to relating the work to its inventor.

Another problem, but usually not a hard one, is the change of the author's style during their lifetime. Some of the researchers suggest that the author's style

changes during his professional career [12]. That is, the development and the enhancement of their styles are related to stage of life in which the work was produced. Also, they consider the form of text they examine because every form has its own statistical properties. That is, the characteristics of prose text or any written language are different than the characteristics of poetry or verse which is organized in a specific way.

1.4 Reasons For Publishing Anonymous Work

One important question must be asked which is what makes an author publish a text anonymously? Or, why does an author wants to hide his personality from the readers? Throughout the research it came up to surface that there are many reasons for this situation to happen. One is political, such as in *the Federalist Papers*, when it was not appropriate for the authors to identify themselves as the authors of the text at publishing time. Another reason is commercial, related as in *The 15th book of Oz*, when the transition of authors of a children story series happened, so the new author could not identify herself straight away because maybe the readers would not like the change. In the case of *A widow and her soldier,* the reason behind what the author did is an attempt to give credibility to the information in the letters, so that readers would not question the information in these letters. Also, the reason in some cases, such as in the *Unabom manifesto,* is criminal reason. The Author Ted Kaczynski didn't want his personality to be known, so as to avoid being arrested by the police.

According to scholars and researchers, the fingerprint of any author cannot be hidden or disguised [11] [12]. An author fingerprint could be revealed by many ways and means, because every writer has his own way of using the words, vocabulary, phrases and sentence construction. Every author has different social background, educational level, and religion that is reflected in his text.

1.5 Text Features and Authors' Fingerprints

Text features are the only aspect that leads to the authors' fingerprints. And, in turn, the fingerprints are the aim that almost every researcher in the field of Authorship Attribution tries to reveal or to discover to solve the ambiguousness of any disputed texts. Throughout the beginning of the authorship attribution field, many statistical and probabilistic approaches took place as a start of finding features such as in [36],[37],[38]. Most of the helpful features are already known to researchers in the field, but the important issue is how to use these features in the correct way to extract useful fingerprints out of the text. There are many features found in text, which help in the area of Authorship Attribution. These features are not something new added to any language, or hidden signs in the text, but they are the language itself. Text features are found in the choice of language grammar constructs, vocabulary, punctuation, words, phrases and sentences. Each one of these is a general aspect that leads to many sub-aspects which lead to distinct features in the end. The usage of such features depends on the author's style and manner of writing.

Every author uses certain vocabulary, invariably different from any other author. One can find that the average of word length for one author tends to differ from that of another author [13] [14]. The number of words in the sentences [44], if they are many or few, depends on the author's style. The construction of the sentence itself and grammar usage mostly rely on the author educational background. Therefore, every writer, author or poet has his own way of feature usage which is his distinct fingerprint, which is very rarely quite close to another writer's fingerprint. Maybe, two authors are close in the pattern of usage of one or two features, but it is unlikely that they are very similar in a large set of features.

Every scholar and researcher uses the features he thinks will reflect and reveal the anonymous or disputed author fingerprint. Some researchers in their studies use features at the character level [1]**[4]**, in which features are groups of (e.g.) two, three or four letters together; this type of feature is often called "n-grams". Other studies depend mainly on word level features (as is the case in this thesis). This usually means that a group of specific words are selected as features to be used. Tracing the frequencies of this set of words links the writer's style to the frequencies of these words. As we will see, the words are usually "function words", and this

method is considered very good as it has been shown that function words as features can really reflect the style of writers in many cases [3],[5],[7],[8],[10],[15],[16],[29] and [32]. Other features sometimes used are word length and sentence length [13][14][44]. Another is vocabulary distribution, which reflects the variety and diversity of vocabulary used by the author.

Donald Foster, an English Professor at Vassar College in New York, and a well-known text Analyst, describes in *Author Unknown book* [11] many of the cases he was involved in as an expert, covering both literary studies and criminal cases. His life as an English professor made him read, study and analyse enormous number of texts of many types, which enlarged his experience in text analysis and investigation. In *Author Unknown* he explained how the features and fingerprints of one author can be different from that of another author depending on many reasons. He says that “there are no two individuals who write in exactly the same way, using the same words in the same combinations, or with the same patterns of spelling and punctuation. No two adults in the same family have read the same books. No one consistently writes fluent sentences. It is that pattern of difference in each writer’s use of language, and the repetition of distinguishing traits, that make it possible for a text analyst to discover the authorship of anonymous or forged documents”.

According to this point of view, that every author has his own style and technique, there is much research to find new type of features that can be used to extract styles and techniques. Vocabulary, grammar, punctuation, word-level and character-level, as we have mentioned, are all kinds of features that can be used to expose authors’ fingerprints. Also, “the language and writing habits reflects the author’s age, religion, education, job, motivation or ideology”. Foster believes that “the author could disguise his appearance, job or anything else but it is not easy for any writer to change his basic vocabulary or his personal store of available words”. Human beings are the "prisoners of their own language. It is not the just the words that a writer uses but the way in which those words are used that makes it possible to distinguish one writer from another."

Another aspect in which text acts like an author mirror is nationality. The Arabic language, for example is spoken in 25 different countries. Every country uses a different vocabulary and even different words for the same thing. Greetings, pronunciations and addressing people are different from one culture to another, even

though all of them speak the same language. All the countries or even areas inside one country use local varieties of spoken Arabic. Still the written Arabic is the same, but this regional language is reflected in the text. Therefore, the nationality of a writer could be identified by his writing and vocabulary even if he writes in the same language spoken in many other countries. Moreover, distinguishing between a foreigner like me and an English native speaker is absolutely clear and obvious.

Foster states that “there are no two writers have identical skills and preferences. In his literary study of any test case he considers the manner in which the quotation marks, carets, cross-outs, dashes and ellipses are written or typed. Even the usage of hyphens, commas periods, colon, semicolons, slashes, spacing and capitalization are important and different from one author to another”.

There are some studies that work on the punctuation as features. Some of these studies count the frequency of every punctuation mark and find patterns of authors with it. Others look for the error in the punctuation marks and others classify punctuation marks syntactically (Using groups of common textual features for authorship attribution).

Spelling plays a great role to differentiate between authors. Foster in his book says that “significant spellings may include acceptable variants that indicate the personal preference (email or e-mail, ensure or insure, skilful or skillful) or regional convention (color or colour, theatre or theatre). Misspelling may indicate dyslexia, or simple ignorance, or deliberate error”. On the other hand, correct spelling may indicate the author’s level of skill. This could help if the text is hand written or not proofread by specialists as in the production of books or novels because it will not change in the style or the spelling.

Grammar is another aspect to distinguish between authors because “grammatical evidence may include pronoun errors (case, number, agreement), or consistency and correctness of verb tenses and auxiliary verbs, or the manners of using comparatives and superlatives, or strictly correct grammar”.

“No two people assemble words or sentences in precisely the same way” according to Foster. “Most writers are largely unaware of their own stylistic preferences, giving no conscious thought to the position of their adverbs or to the frequency of their use of, for example, the passive voice. The writer’s syntax will usually remain fairly constant from one type of writing to another. For example, any

writer cannot suddenly abandon his manner of connecting clauses. The writer must use the language he knows, it is arranged and formed from his own bank of words".

Foster's views are much in support of the idea of using "function words" as features. Function words (like "and", "but", "so", and so on) are words that authors use unconsciously according to his manner of writing. He cannot change it according to the nature of the topic or according to the subject being written about. Function words include many prepositions and conjunctions that are used independently and are unrelated to the topic.

In a classic study by John Livingston Lowes in his book *the Road to Xanadu*, he investigates that "any writer or author keeps in his mind unconsciously some of the words, phrases and sentences he had read or listened to before. Then, the writer uses these words or phrases in his writing without thinking. This would help in constructing an image about any writer's previous experience in reading and learning". Therefore, authors not only have their special style which reflects their fingerprints but also they use words, and sentences they gained from the past, which in turn reflects their knowledge background.

1.6 Arabic Language in Authorship Attribution

Arabic language is one of the most spoken languages in the world. It is considered the fifth most spoken language. Its origin or as known historically is from Saudi Arabia and Arabia Area. It belongs to the Afro-Asiatic Language family as explained in details in chapter 4. Arabic is spoken in the Arab world as an official language in 22 countries. The number of people who speaks Arabic in the Arab world is around 340 million people. So it is widely spread and spoken by big number of people.

Authorship attribution research in Arabic language is considered new and is not tackled as much as in other languages. In the preparation for this research, there was no any published works and researches about authorship attribution in Arabic language. This encouraged the idea of starting and establishes a research in Arabic authorship Attribution.

There are very interesting issues in the Arabic language that attracts every Arabic person and especially Muslim ones. This issue is the quotes of the prophet Mohamed (peace be upon him). Prophet Muhammad is the messenger of god to

people to introduce the religion of Islam. So in Islam there are two main resources that contain all the basics and rules that govern the Islam religion. The first and most important one is the Quran book, which is word of God revealed to Prophet Muhammad by Gabriel the archangel. The other important resource is the Prophet Muhammad's life and traditions, which was the verbal and physical traditions. These traditions are collection of quotes known by (Hadith) that were told by prophet Muhammad to his followers and companions to teach and clarify a lot of aspects in Islam religion. These quotes were memorized and written by many of his followers. These quotes were passed on and conveyed from one generation to another. In the third generation of the prophet companions, many Muslim scholars tried to collect and gather all these quotes and to be put in one book to be preserved. One of the most famous scholars is Called Muhammad ibn Ismail Al-Bukhari. He made a book of around 7500 of the most authentic quotes. He spent 16 years gathering and researching these quotes. He gathered more than 600,000 quote but he only printed in his book 7500 quotes which he believes are the most authentic and true that have the exact same words of the prophet. All the Muslim scholars agreed that this book is the most authentic book of quotes.

These quotes are important for every Muslim because it explains many of the important issues in Islam. So the idea of studying and searching to find a way to differentiate between the authentic and true quotes of Prophet Muhammad from the added or not completely true ones is considered extremely motivating.

As mentioned before this research is considered the first in Arabic Authorship attribution so finding text features and classification tool that can discriminate between authors is the primary objective and goal of the thesis to start with. Then after establishing firm ground in the Arabic Authorship Attribution, one could study the problem of Prophet Muhammad quotes as future target.

The finding of electronic texts in Arabic language is considered a problem that faced the research in the beginning. In a report by The National news press [45] this problem is a result of that there are no big firms that support development of software that supports electronic versions of Arabic books. The main problem is the lack of conversion of Arabic language into electronic format. Moreover, there are no real platforms and that supports Arabic Language contents on any of the major nowadays tablets such as Amazon Kindle, e-readers, i-Pad and any electronic

devices. So the production of Arabic electronic texts is very scarce. There are few e-books and the majority is in Poor PDF format.

The Arabic book publication per year in the Arabic countries as published by the United Nations Educational, Scientific and cultural Organization (UNISCO) is about 21,450 books. It is considered very little number in comparison of the number of Arab countries and the number of peoples who speaks Arabic. This small number of publications is related to the high percentage of illiteracy in the Arab countries. But on the other hand, there are promising countries such as Egypt, Lebanon and Saudi Arabia who are lately increased their publications. According to Egyptian Central Agency of Public mobilization and statistics that the latest statistics about published books number is in increase. The numbers since year 2005 was 2515 newly published book and 2953 in year 2008. This reflects the increase of publication number between these years.

1.7 The Organization Of The Thesis

The thesis consists of Seven chapters. Chapter one is an introductory chapter about the authorship attribution field. It describes the different types of authorship attribution problems and the obstacles that face the researcher in that field. It also describes text features and how they reflect authors' fingerprints.

Chapter two is a literature review of the authorship attribution field. It shows the different approaches used in this field and the classification tools that are used in researches and experiments. Also, illustrations of different types of features that can be employed in experiments stated in this chapter. Chapter two describes the material and data that many tests are carried out on.

In Chapter three, experiments in the English language, I present the research work using English function works to classify the federalist papers and the 15th book of oz. This was a study that focussed on two types of feature selection method, and one of these continues to be used in later chapters.

Chapter four, processing Arabic text, talks about how the Arabic material used in this research is found and collected. Also, it shows the processing and editing of these datasets to make them ready for the experiments. As well, this chapter includes a description of the software implemented to calculate the frequency of

function words and characters in text. Then, the difference between Arabic and English language is explained and illustrated by trees of origin.

Chapter five is considered the main contribution of the thesis, demonstrating the selection and successful use of function words for authorship attribution in Arabic. It shows and describes the Arabic function words and the words chunk sizes used in the experiments. Moreover, it explains the classification tool which is used in the experiments and shows the function and describes the evolutionary algorithm used to select features. Then, experiments and results are clearly stated and presented.

Chapter six shows the experiments done in the Arabic language with another type of feature. Character-level unigram features are described in general and used in experiments to differentiate authors. And also, this chapter is about enhancing the performance in Arabic by combining different types of features, such as function words and Character Level. Also, it shows another attempt of enhancement by combining two different distributions of function words.

Chapter Seven presents the conclusion to the thesis and suggests future work.

1.8 Contributions

Contributions are numbered C1--C5; this numbering is useful when we refer to them later in the concluding chapter.

- C1: The identification of function words for the Arabic Language that are successful in the Authorship Attribution problem Studies. The identification and use of function words in Arabic is here considered to be used for the first time in this field.

- C2: The use of two Hybrid Algorithms (Evolutionary Algorithm / Linear Discriminant Analysis) were successful in the Authorship Attribution problem in both English and Arabic, especially in terms of finding the least numbers of features to solve the problem.

- C3: Investigation and assessment of character level features in Authorship Attribution problem in Arabic Language. Again, this is considered to be the first such study.
- C4: Contribution several datasets to be used by other studies in the field of Authorship Attribution in Arabic Language. The datasets are edited and available by myself on request.
- C5: Contributing a number of new approaches to Authorship Attribution Studies by combining different chunk sizes or different distributions of function words frequencies. Also, combining the function words and character level features together in an attempt to enhance performance.

1.9 Originality Of The Thesis

The thesis is mainly about investigating the Authorship Attribution problem in the Arabic language, and how it is compared to Authorship Attribution in English Language. This is considered one of the first researches to be done in the Arabic Language in the Authorship Attribution field using Arabic Function Words. A hybrid algorithm (Evolutionary Algorithm /Linear Discriminant Analysis) is used to reach the best least number of features to classify the problem. The features used in the research are a list of Arabic Function words which are the first such list to be published.

2

Literature Review

Chapter overview

.In this chapter I provide a Literature review of the Authorship Attribution problem. In section 2.1 is the introduction to this literature review. Section 2.2 is about the Authorship Attribution problem and describes some examples. In this section I introduce the most famous and well-known examples that have appeared in the field. Section 2.3 is about the classification tools that tend to be used in the Authorship Attribution area. Section 2.4 is about the different types of features of text that are used for this problem. Section 2.5 is about Authorship Attribution in different languages. Section 2.5 is the conclusion to this review.

2.1 Introduction

The main aim of the thesis is to investigate ways to solve the authorship attribution problem in the Arabic Language. To attain this goal or target first one should try to find the most appropriate tools and text features that can be used to classify disputed works in the Arabic language. A good start for this is to consider the tools and features that have succeeded in English and other languages.

There are three main components that play a great role in the Authorship Attribution field. These components are feature extraction, classification tools, and datasets. These three components must be used all together wisely in finding ways for solving the Authorship Attribution problem. Feature extraction is the most important component because, with the right choices of features, it is hoped that the datasets will reflect the writer's style to a great extent, which helps in differentiating authors by their works. Feature extraction is considered as searching for the fingerprint of the author. The classification tool comes in second place, as it classifies the authorship of the disputed works to one author or another. It is important to find a suitable classification tool that suits the nature and the dimensions of the problem. The dataset is important because it is the material or medium upon which you apply your feature and classification tool. It is important to find good material which is genuine and original and that reflects the authors who wrote it. Meanwhile, when methods are being developed and tested, it is important that the datasets for testing are varied. For example they should not be all easy problems to solve. I am concerned to find the best features that work well perfectly with a classification tool to reach the best accuracy level in classifying disputed works.

The literature review is divided into four main sections. The first section is introducing the most famous and popular Authorship Attribution problems and examples. The second section is about the computational methods and tools used to solve these problems. The third section is about the features used in many researches and experiments. The fourth section is about Authorship Attribution in different languages.

2.2 Authorship Attribution Problems and Examples

In the area of Authorship Attribution, there are famous cases and examples which many researchers use in their studies and experiments. There are papers and researches performed surveys on authorship attribution [13][34]. As a matter of fact, the **Federalist Papers** are the most famous case in this field. They are 85 essays published anonymously in newspapers in 1787 and 1788 to persuade the people of New York to approve the new American Law. The authors of the essays were Alexander Hamilton, James Madison and John Lay. Hamilton and Madison each claimed to be the author of 11 of the essays. Several papers in authorship attribution have used the Federalist papers as their experimental dataset. In the paper by Khmelev and Tweedie [1] used Markov chains of the measured probabilities of subsequent letters. That is they turned every text into a matrix of probabilities of subsequent letters of the English alphabet. All punctuation and formatting are removed. They used 387 texts of forty five authors from project Gutenberg archives, data from Halteren, Tweedie and Baayen [31] and the Federalist Papers. In the texts from project Gutenberg, 74.42% accuracy was achieved and they assigned the disputed papers in the federalist papers case correctly. In the paper by Fung [3], he used the method proposed by Bradley and Mangasarian which is Feature selection via concave Minimization and Support Vector Machines. The text used was the Federalist Papers problem. They used the relative frequencies of the 70 words, which were identified by Mostaller and Wallace [15], per 1000 words. They found a separating plane that classifies correctly all the training dataset of known authors using the relative frequencies of three words (To, Upon and Would). They classified all the disputed papers using the separating hyperplane in three dimensions. In another paper by B. Kjell [35], the author used the neural network and relative frequency of letter pair as text feature.

In the paper by Oakes [9] used the Ant Colony Optimisation to solve the disputed authorship in the case of Federalist Papers. He produced rules in the form of IF <condition> THEN <Class> where conditions are sequence of terms that are compared using AND operator. The terms used were word frequencies of 30 markers words identified by Mostallar and Wallace [15]. The rules are discovered from the training data then applied to the test data. He achieved classification accuracy of 79.1%. Each one of these papers used different classification tools and different

features. The usage of one dataset in different papers is beneficial for research because in this case one can compare the performance of the different methods and features used by the authors. For example in [3], the authors used the Federalist Papers as a dataset and Support Vector Machines as the classification tool, and function words as the features. I found this paper suitable to compare with my own work on the Federalist papers and using function words as features. Instead of the Support Vector Machine, in that work we used Linear Discriminant Analysis and we achieved better results according to the number of function words used in classification of the disputed papers (this is described in chapter 3). Some papers such as [39] used the federalist papers as a testing grounds and applied three different features to compare their performances.

The **15th book of OZ** also is considered one of the most famous disputed cases in English literature. The Oz books are series of books for children which began in 1900 with *The Wonderful Wizard of Oz*. The OZ tales were started by Lyman Frank Baum (1856- 1919) who wrote the last tale which was the *Glinda of Oz*, which is the 14th book of OZ tales, then he died in 1919. Then, an established children's writer Ruth Plumly Thompson (1891-1976) completed the work Baum started and she produced another 19 tales of OZ. After the death of Baum by three years, the 15th book of OZ tales was published in 1921 by the name of *THE ROYAL BOOK OF OZ* and Baum's name was on the cover and Thompson was only acknowledged only as having "enlarged and edited" the work. Later, Thompson was acknowledged as having written the work. In paper by Binongo [10] tried to solve the problem of the 15th book of oz authorship problem. He used the 14 books of Baum's and 14 books of Thompson as the dataset. He used the most frequent 50 function words in the 28 books. He found the average rate of each word in 5000 words chunks. Then he constructed a matrix of 50 words x 223 text blocks. Then he reduced the 50 dimensions into 2-dimensions to be visualized using the Principal Component Analysis. He showed that it can be classify the disputed book graphically using the Principal component Analysis. The author classified the disputed book as Thompson work.

The **Pickett letters** texts story is that in 1908 the widow of Gorge E. Pickett published forty-four letters that were written to her by her husband during the four years of American civil war under the name of *A Widow and her Soldier* and also

under the name *The Heart of a soldier*. Later, some historians suspected that the author of these letters could not be General Pickett himself, but his wife because there are some facts about the civil war in the letters that they think he would not have written. In paper by Holmes, Gordon and Wilson [8] collected many texts of the same genre as control texts. They also gathered genuine material for George Pickett in the form of military reports published in official records of US war department. On the other hand, material collected for Pickett's widow from novels she published and original personal letters she wrote. Principal component analysis and descriminant functions were used in this research as the classification tool. Features used in the research were top sixty function words frequencies per thousands word when using the Principal component analysis and the top 25 function words frequencies when using the discriminanat functions. The research suggests that La sale Corbell is considered the composer and the author of the published letters not the George Pickett's original letters except for 11 letters.

Many other researchers investigate the authorship attribution problem without using texts from famous papers. They use 'ordinary' material to validate their techniques. For example, Sanderson and Guenter [2] used text as dataset from 50 newspaper journalist with a minimum of 10000 words per journalist. The aim of the research is to evaluate the usefulness of the sequence based approaches in solving authorship attribution problem and to compare their performance with probabilistic approaches based on Markov chains of character and words. They found that bag-of-words kernel performed better than word sequence Kernel and Character sequence kernels is better than bag-of-words kernel. And in paper [6] Stamatatos, Fakotakis and Kokkinakis used downloaded texts from Modern Weekly newspaper which is Greek newspaper. They extracted the style markers from the texts using a special tool called sentence and boundaries detector. It segments the input text into sentences and then detects the boundaries of intra sentential phrases. It produces 22 style markers for each text. They used multiple regression in predicting the authors of the texts.

Authors Morales, Pineda, Gomez and Rosso [5] gathered and collected 353 poems written by five Mexican poets. These poems are considered very short texts as the average number of words is 176 words. They used the Naive Bayes classifier as the classification tool because it suits the idea of Maximal Frequent word Sequence

which they used to extract features of the texts. The features extracted were set of relevant sequences that combine functional and content words. They did experiments with different kinds of features types. They used function words, content words, combination of function and content words and n-grams words. They found that extracting words sequences by frequency of occurrence is better than by length of the word sequences. The overall results achieved 83% accuracy. While in Baayen, Halteren, Neijt and Tweedie [7] they used in their experiment texts which are written by authors with very similar background and training. Moreover, the genre and topic were strictly controlled too. The main aim of the research is to test the existence of author's fingerprint and if it could be extracted and identified even the authors have similar background and training. The authors of the texts are eight Dutch literature students. Every student wrote a total of 9 texts in three different genres, 3 texts each. The genres were fiction, argument and description. 50 function words and 8 punctuation marks were used as the features. Linear discriminant analysis was the classification tool. They achieved 88.1% accuracy which suggests that every author has a textual fingerprint even they have similar background and training.

The book *Author unknown* [11] contains more than five cases that the author Don Foster himself investigated, but without using any specialised computational methods. He uses all he can extract from the text as features to help him solve cases; he examines words and punctuation, looks for indentations and split infinitives, and generally considers all parts of speech. He looks into every aspect of the language that may help to distinguish one writer from another. What makes me interested to know how he works is that his way of dealing with text can lead to the discovery of many features that can be used by researchers in experiments to extract authors' fingerprints. He explains his method of classifying anonymous text by the following steps:

First, as a primary step he tries to find similarities between anonymous text and text databases. This step doesn't help solve the problem but "it gives indications about the author's age, religion, education, job, motivation or ideology". He adds that "the study of an anonymous text doesn't produce decisive authorship attribution, but he can narrow the field the field of suspects by isolating the geographic, ethnic, socioeconomic, corporate or professional context to which the unknown writer belongs".

Second, he looks for "familiar words misused in sentences". He thinks that "it is not the words that a writer uses but the way in which those words are used or abused that makes it possible to distinguish one writer from another".

Third, Foster examines orthography and punctuation. He considers "the quotation marks, carets, cross-outs, dashes, and ellipsis, either written or typed". He looks at "handwritten symbols, dollar signs and ampersands". He also, looks at "the use and omission of periods with abbreviation and acronyms". He looks at "the writer's use of hyphens, commas, periods, colons, semicolons, slashes, spacing and capitalization".

Fourth, spelling reflects many characteristic of the author's identity. As a text specialist, he looks for spelling mistakes that give clues of the writer's habits, level of skill, or details of the word processor used for spell-checking. This also indicates if the writer is a native speaker of the language or not.

Fifth, he looks at grammatical evidence such as "pronoun errors, or consistency and correctness of verb tense and auxiliary verbs, or the manner of using comparatives and superlatives; almost any repeated and characteristics lapse, or even strictly correct grammar", which can be used to collect a lot of characteristics about the author. He adds that "the author's first or native language could influence the grammar structure of the language he uses".

Finally, and the most important point that supports the features used in thesis, is that he looks at how the author puts words together. He believes that "no two authors assemble words or sentences in precisely the same way. Most writers, professional writers, even professors of English and linguistics, are largely unconscious or oblivious to their own stylistic preferences, giving no conscious thought to the position of their adverbs or to the frequency of their use of the passive voice". The type of document being written could change from essay to letter or to any kind, but the author does not suddenly change his preferences and his style, for example, for words used for connecting clauses.

The five cases that were in the book *Author unknown* are as follows:

Funeral Elegy is a funeral poem written for a person called William Peter and dedicated to his elder brother John. It was printed in 1612. The poem was only signed by the initials W.S. The unusual thing about this poem is that it was signed

W.S. twice, in the beginning and at the end of the poem. There was nothing else indicating the author.

The next case is *Primary colors*. In January 16, 1996 a novel called *Primary Colors* was published anonymously. The story is about the United States of America presidential campaign of Bill Clinton in 1992. The novel hit sales records because of the fact that it was anonymous and many speculations took place about the true writer of this novel

The Unabomber manifesto is a group of letters and a 35,000 words essay called *Industrial society and its future*, published anonymously in 1995 in The New York Times. Later the manifesto was acknowledged as being by a university academic called Theodore John Kaczynski. In this manifesto he declared and stated his goals and urged to be met in order to stop mailing bombs and explosives to university professors and businessmen. The aim of publishing this manifesto was to try if anyone could identify the author of it so that the FBI could arrest him. Kaczynski's sister in-law identified him by his ideas from the text and the expressions he uses when she read the manifesto. During the court case the FBI wanted to make sure that the manifesto was authored by Kaczynski, so they invited Don Foster to examine it.

The Letters of Wanda Tinasky were a collection of comic and playful letters sent to the Anderson Valley Advertiser by a pseudonymous Wanda Tinasky between 1983 and 1988. These letters were collected and published later without the finding the real author of those letters.

The final case is the controversy about the disputed poem "Twas the night before Christmas" or "A visit from St. Nicolas", which is considered to be the origin of our modern image of Santa Clause and his reindeers. It was published anonymously in December 1923 by a New York newspaper, the Sentinel. Then, in 1944 it was declared as a poem by Clement Clarke Moore, as he included it in his book of poetry. The story behind the anonymous publication of the poem is that a family friend sent the poem to a newspaper one year after she listened to Clement Clark Moore was reading it to his wife and children. He wrote many poems for his children and explained that his reason for publishing these poems anonymously was because he believed his work in translation is more important and he didn't want his image to be related to his children's poems but to be related to his work in Hebrew

translation. Later, some scholars suggested that the actual writer of the poem is Major Henry Livingstone. Don Foster believes that Clement Clark Moore is not the original writer of the poem. He justifies his beliefs by some evidence. He suggests that “the style and manner of the poem is completely different than all of the other works of Clement Clarke Moore”. And he stated that “the way Major Livingstone writes and his approach to poems totally agrees with the disputed poem”.

2.3 Classification Tools

There are many classification tools used in the area of Authorship Attribution. Some of these tools have been shown to achieve high performance and accuracy. In other cases in which classification methods are found not to work well, it is often argued that the problems studied in these cases are too complicated to solve clearly. The authors in [1] use an approach based on Markov chains to provide a general technique in the authorship attribution area. The authors built a first-order Markov chain transition matrix, representing for every pair of English characters, the probability that the first will follow the second, and used it on three different datasets. In addition, this approach was compared in [2] to a Support Vector Machine classifier, and it was found to perform better in some of the cases.

Support Vector Machines have been tried several times in the Authorship Attribution field. As indicated above, in [2] it was found to perform well in comparison with Markov Chain in some cases, and in [3] this approach achieved good results and classified all the disputed works correctly using only a small number of features. In the study by A. Shlomo and L Shlomo [3] SMO which is a support vector machine algorithm performed very good achieving 99% in discrimination using function words and the next best result was 94 using frequent collocations of window size 10. In a study [40] that introduced evolutionary algorithm that uses support vector machine and feature selection algorithm. It achieved the highest accuracy of 97-99% in English and 92-93% in Chinese language. However the Support Vector Machine approach has its drawbacks; it is much more complicated than the Markov Chain approach, and most other approaches, in a way that makes it difficult to use (e.g. to tailor and configure for a particular instance) for someone who is non-expert in mathematical and statistical machine learning. Moreover, the Support Vector Machine approach can only

naturally apply to the separation between two classes; although techniques exist that exploit Support Vector Machines in a way that allows multi-class classification (this is the situation in several cases in authorship attribution), these contain extra complexities and parameters that make them difficult to use

Linear Discriminant Analysis was used [7] and was found to perform well on datasets where researchers tried to classify topics from authors of the same age, background and genre. Also, when used in [8] a linear discriminant analysis approach involving two derived discriminant functions managed high accuracy classification in all the test datasets, especially after modifying the input of the Linear Discriminant Analysis by appropriately weighting the input vectors. The final accuracy achieved in this paper reached 88.1%.

Principal Component Analysis is often treated as a graphical and exploratory analysis tool more than as a classification tool. It is mainly used to show to what extent different classes are separable. Eg, Principal Component Analysis [8] was used to validate the features used in that study and to show if these feature can separate between authors. But in [10], Principal Component Analysis was used as the chief classification tool. The authors succeeded in separating the two different classes and aligned the disputed papers clearly and perfectly using Principal Component Analysis.

Bayesian Decision Theory was used in a classification problem involving three different languages in [4]. Peng, Schuurmans, Keselj and Wang used three different languages in their experiments. These languages are Greek, English and Chinese. The Greek data used are the same as in paper [6] which is downloaded texts from Modern Weekly newspaper. They divided the data into two sets. Every set consists of 200 documents written by 10 different authors. The first set which is called set A is for authors that write on variety of topics. In set B are for scholars that write topics about science. The main difference between the two set is that set A is heterogeneous and set B is homogeneous. For the English language, the data used is collected from Alex Catalogue of Electronic Text. They used 8 of the most prolific authors from this collection such as Charles Dickens and William Shakespeare. For the Chinese corpus, they downloaded Eight of the most popular modern Chinese martial Art novelists. The main idea of the research is to build a character level n-

gram language model for every author. They are not selecting features but they use all the features in the model. The approach used in applying the Language models to authorship attribution is by using Bayesian decision theory. The best result for the greek language achieved was 74% for set A and 90% for set B. In English Language, they achieved the best accuracy which is 98% using 6-gram model . In Chinese Language, 93% accuracy is achieved.

The Naïve Bayes classifier, which is a simple probabilistic classifier, uses the method of maximum likelihood, and was used in [5] on Mexican language data to classify short texts of an average size of 176 words per text. The accuracy level of this technique reached 83%. However this accuracy level is not bad on such small sized texts. In [8] it is stated that an optimum or sufficient size is about 1000 words, while in [10] the authors used about 5000 words per text.

Ant Colony Optimisation, which is used in [9], is a swarm intelligence algorithm, usually employed to solve optimisation problems; as with many other optimisation algorithms, it is also used often for machine learning and classification—i.e. it is used to find a model that optimizes accuracy on a training dataset. The study in [9] used ant colony optimisation to find a classification model in the form of logical rules with operators. The rules are generated from the training data and then applied to the test data. The average accuracy was 79%. However this did not perform well in comparison to other tools and methods used in other research papers such as [5] which achieved 83% accuracy, [8] which achieved 88.1% and [4] which achieved 98% in English language.

The accuracy levels obtained in these studies using these different classification methods does not really reflect the goodness or badness of the classifiers. This is because there are many other different elements that affect the classification in these cases, such as the features used and the datasets used in experiments. If we can develop a number of standard datasets, as has been done in many other areas, then it will be possible to progress much more rapidly in the field of authorship attribution.

2.4 Features

Features play a great role in authorship attribution. Much research in this area aims to try to find the features that best reflect the author's style identity or his fingerprint. Also, most features are considered language-dependent because a feature that may work in one language may not work for another. That is because not all languages have the same origin or nature, which in turn underpins different grammar conventions. Many studies have been done that lead to representing the world's different languages in a complex family tree. When two languages are close in this tree, we might suggest that the same features could work well in both these languages. For example, Danish, Swedish and Norwegian share the same branch in the part of the tree that is Northeast Germanic Languages. These languages share about 60% to 70% of their vocabulary. Another example is languages that belong to the Romance family, again sharing a lot of similarities in grammar and vocabulary, such as French, Italian, Spanish and Portuguese. On the other hand, languages that belong to different family subtrees share few characteristics. Arabic, which belongs to the Afro-Asiatic Languages subtree, is totally different in structure and grammar from English, which belongs to the Indo-European Family.

The concept of *function word*s as features is the most popular source of features in the area of authorship attribution. As we saw first in Chapter 1, and will return to later, function words are connecting words (such as "and", "then", "whereas", etc.) that are independent of the content or topic, and which seem to be used in different ways and patterns by different authors. The majority of studies use function words as their features for several reasons. First, function words are the earliest-used features that performed well in this field, as pioneered by Mosteller and Wallace [15]. Second, function words make good intuitive sense, since it is believed that authors do not consciously vary or control the usage of this feature in their writing. That is, the writer uses function words unintentionally, in ways that to a great extent reflects his style and fingerprint. In a great psychological study of function words by C. Chung and J. Pennebaker called *The Psychological Functions of the function words* [33]. The study is mainly about how function words use reflects the social, personality, cognitive and biological background. They used text analysis program called Linguistic Inquiry and word count. They have done their analyses on 95,000 text files representing over 80,000 different persons. They stated

in the study that ”the average native English speaker has an impressive vocabulary of well over 100,000 words, fewer than 400 are function words. This deceptively trivial percentage (less than 0.04%) of our vocabulary accounts for over half the words we use in daily speech.” Just the 10 most common used function words used as much as 20% of the words we use every day. They added “In daily conversations, however, we have virtually no control or memory over how and when they are used either by speaker or by ourselves. Despite rarely paying them any conscious attention, function words have powerful impact on the listener/reader and at ther same time, reflect a great deal about the speaker/writer.” Also, as stated by A. Shlomo and L. Shlomo in their paper [32],”due to their high frequency in the language and highly grammatical roles, function words are very unlikely to be subject to conscious control by the author. At the same time, the frequencies of different function words vary greatly across different authors and genres of text, hence the expectation that modelling the interdependence of different function word frequencies with style will result in effective attribution.”

Third, there is much evidence that they work well as features, and are widely known and understood in the authorship attribution field. For example, the authors [3] succeeded to reach high accuracy in authorship attribution using only three function words, and classifying using a Support Vector Machine classification tool. Also, function words were used in [8] in connection with Discriminant Analysis, and were found to lead to excellent results. In [10] (as well as in [8]), principal component analysis based on function words features worked well at separating the data from separate authors. Function words in conjunction with punctuation marks were investigated as features in [7], and this combination was found to perform better than the usage of functions words alone. Moreover, in [5], function words were used in combination with off-content words. In a study [32], function words were compared to other features such as frequent pairs and collocations, the results supported that function words gave the highest overall discrimination in comparison to other features. In this study the classification tool was SMO which is support vector machine algorithm. The dataset used was a collection of 20 novels. Each novel was divided into chapters, which gave a total of 633 chapters. The papers recently mentioned, along with many authors, investigate author attribution using

function words as the primary source of features. This reflects the success of function words in this field.

There is therefore much evidence that function words are really helpful and beneficial features in the field of authorship attribution. Three of the papers that we have referred to which used function words were doing studies in the English language, one involved Dutch and the other involved Spanish. Dutch and English language come from the same natural language sub-tree; it is demonstrable that the similarities between languages as close together as English and Dutch is reflected in the similarity between the nature and use of function words in the two languages, and so it is not surprising that function words features work well in both cases. The general success attained by authorship attribution studies based on function words features encouraged me to look for the equivalent, if possible, of function words in the Arabic language. However, since the Arabic Language is quite different in origin and nature from the many languages that have appeared so far in authorship attribution studies using function words, it was initially unclear or doubtful if Arabic analogues of function words would achieve the same level of success.

An alternative and also popular approach is to use character-level features. That is, instead of patterns in the use of certain words, we look for patterns in the sequencing of characters. Character level features are promising in that they can be used straightforwardly in more than one language. This is because it does not depend on aspects of the grammar or nature of the language, except for the fact that the written form of the language must be a linear string of symbols from a fixed alphabet. In character level studies, features based on sequences and combinations of characters can be constructed in many different ways to be used in classification. In [1], for example, the authors used the probability of subsequent letters as a feature. That is, an author's style fingerprint is represented as the set of frequencies of letter pairs in his or her text. For example, they count how many times letter 'A' occurred in followed by letter 'B', then how many time in followed by letter 'C' and so on. This is calculated from an author's works. Then a transition matrix of first characters (as a column) and second characters (as a row) is constructed. As they include all the alphabet letters, they end with 26 characters plus the space character to form 27 characters. and then stored in a 27×27 transition matrix of probabilities of all characters. A transitional matrix is then calculated for each text. In this context, a

sequence of *n* characters is called an *n*-gram. That is, a sequence of two letters is called 2-gram or bi-gram, and a sequence of three letters is called 3-gram or tri-gram. Studies which use the character level n-gram techniques select certain number of sequenced letters to use as a feature, such as 3 letters sequences or more. In [2] and [4], texts were characterised by *n*-grams where *n* was as much as 6 (6 letters). In [35] Author used the relative frequencies of the letter pairs as a text feature using neural network.

Character-level features, as we can see, are sometimes used in authorship attribution and many researchers find the classification performance acceptable. Since it is straightforward to investigate in any new language (as long as the language is fundamentally character-based), it is a candidate for trying in the context of Arabic texts.

The use of word-level features in authorship attribution has already been discussion in the context of function words. However there are alternative approaches that use word-level features (i.e. not restricted to function words). As one of the earliest work on world level using word length, in 1887, Mendenhall [41] did research using word-length distribution between prose and poetry not on different authors [43], on some works of John sturart Mill and then applied the same feature on Shakespeare and Bacon[42]. Also, for example, in [2] the features are based on the probabilities of word sequences; with classification done by Support Vector Machines, it was found that strong performance was achieved with sequences of length 4 words. Also, word sequences were used in [5], including both function words and content words, and using a Naïve Bayes Classifier.

Another category of feature is the so-called ’style marker’. In [6], 22 style markers were used as features: these comprised 3 token-level markers. The token level markers are sentence count, word count, punctuation marks count, etc. 10 phrase-level markers, which are noun phrase count, word included in noun phrases count, prepositional phrase count, word included in propositional phrase count, etc. And 9 analysis-level markers, which are unanalyzed word count after each parse, non-matching word count and assigned morphological descriptions for both words and chunks. Style-markers were also used in [9], but the results (using classification models optimised with ant colony optimisation) were not impressive. It seems

reasonable to conclude that style markers are quite language dependent, and the appropriate choices of style marker features will differ greatly from one language to another. Punctuation, for example, is much more rigid in English than in Arabic. In English punctuation can play a critical role in clarifying the meaning of the sentence; but in languages such as Arabic, there are no specific rules that govern the usage of punctuation marks. In some ways this could suggest that punctuation based markers may be more useful as features in Arabic authorship attribution than in English. However, the point being made, in the current context, is that the value of a given style marker feature in one language may be very different to its usefulness in another language. Moreover, punctuation marks are different from one language to another. Spanish language uses inverted question marks "¿" before the question, and regular ones after the question. In Arabic and Persian, the comma (،), question mark (؟) and semicolon (؛) are reversed because of the nature of writing from right to left. In the Greek language, a question mark is written like the English semicolon.

Function words and character level show that they are considered from the best features, which really reflects the author style and identity, according to the results achieved when used then in the research papers. Character Level is much promising in the area of general usage because of its language independency. While Function Words is considered a very successful feature in English and in some other languages that shares or relates to English language origin or nature such as Dutch and Spanish Languages.

2.5 Authorship Attribution in Languages Other Than English

There have been several research efforts investigating authorship attribution in languages other than English language. In searching the literature I found authorship attribution studies in a total of seven different languages as shown in table 2-1. Arabic was not included among these languages, so I feel confident to claim that the work described in this thesis (incorporating (Shaker & Corne, 2010)) represents the first study in this area. However there has been research in a related application: in [17], author Eistival is interested in the Arabic language, and the goal is to provide information about the authors of both English and Arabic e-mail messages. The corpus consists of e-mails from 1033 English speakers and 1030 Arabic speakers.

The main goal of the research is to develop a tool that can give information about the author. The system is called Appen TAT. The system takes documents as an input and produce statistical descriptions of author as an output. They try to predict the authors' demographic traits, such as gender, age, geographic origin and level of education. This is a different and intuitively simpler task than (for example) attempting to distinguish between two or more authors that may in fact have the same demographic profiles. The types of features used in [17] which are document and linguistic features are entirely different than the function words in our work. Sentence punctuation marks and other special characters such as emoticons are one type of features used. Part-of-speech tags are other features. Also, the lists of Arabic function words used in the thesis are considered the first and initial function words to be produced for the sake of research in Arabic Language.

Some papers have tried to compare a specific approach across many languages. For example, paper[4] English, Greek and Chinese texts were used in the experiments. Mostly, however, a single research work focuses on only one language. E.g. in [6] the language of interest is Greek, and the work involves a collection of Greek newspaper articles. Spanish was explored in [5] using a dataset of poems. In [7], Dutch was the language of study, and they tested word sequences and punctuation marks as features. In [16], Latin function words and word sequence were used as the features to solve the problem at hand. Finally, in [18] Reicher, Kristo, Belsa and Silic used the Croatian language in their research. They used three different datasets. One dataset consists of 4571 journal articles written by 25 authors downloaded from daily Croatian newspaper. The second dataset consists of 3662 online blogs by 22 authors. The third data set consists of 52 novels by 20 different authors from classic Croatian literature. They used different types of features such as function words, Part-Of-Speech which is the number of occurrences of different tags (adverbs, adverbs, conjunctions, particles, interjections, nouns, verbs, adjectives and pronouns), word Morphological Category, Part-Of-Speech n-grams and other features as simple characters and lexical features. The classification tool used was Support Vector Machine with radial basis functions as the kernel. They stated that function words, punctuation marks, word length and sentence length are the most successful ones in achieving high accuracy.

Paper Number	Material Used	Tool	Feature	Language
Paper [1]	Project Gutenburg Federalist Papers	Markov Chains	Probabilities of subsequent letters	English
Paper [2]	Newspaper Articles	Support Vector Machine Markov Chains	Character Sequence and word Sequence	English
Paper [3]	Federalist Papers	Support Vector Machine	Function word	English
Paper [4]	Different Collection	Bayesian Decision Theory	Character Level	English Greek Chinese
Paper [5]	Poems	Naïve Bayes Classifier	Word sequence	Spanish
Paper [6]	Newspaper Articles	Sentence and chunk Boundaries	Style Marker	Greek
Paper [7]	Topics on 3 genre	Linear Discriminant Analysis	Function Words and punctuation Marks	Dutch
Paper [8]	Pickett Letters	Discriminant Analysis	Function Words	English
Paper [9]	Federalist Papers	Ant Colony Optimisation	Style Marker	English
Paper [10]	15th Book of Oz	Principal Component Analysis	Function Words	English
Paper [16]	De Doctrina Cristiana	Principal Component Analysis	Function words	Latin
Paper [18]	Articles, blogs and Books	Support Vector Machine	Character, Lexical and syntactic Level	Croatian

Table 2-1: This table includes researches and studies done in different Languages. It also shows the classification tool and the features used.

2.6 Methods and Concepts

In this section I would like to introduce some of the main methods and techniques which the work of the thesis depends on.

2.6.1 General problem of supervised learning

What is the supervised learning? It is a machine learning methodology that uses supervised training dataset in which the input is trained to predict a predefined or desired output. Then, as a result of the training process a classifier is produced which is able to predict correct output according to the input vectors given. A proper definition of the supervised learning [46] is "A common task in a machine learning is to learn the function relationship between the input and output. The inputs x are generally vectors of features, which may be discrete, continuous or mixed. The output is typically scalar y, the target. If y is a continuous variable then the problem is known as regression problem. If y is a discrete variable then the problem is known as classification problem, y indicates into which class the observation x falls. During the supervised learning the machine is equipped by a set of training data comprising pairs $\{x_n, y_n\}_{n=1}^{N}$ of features and targets which are assumed to be representative of the process being modelled. If the $x \rightarrow y$ is successfully learned, then the learned function can be used to make prediction of the target for feature whose target is unknown."

2.6.2 Linear Discriminant Analysis

The general definition of the linear discriminant analysis is a technique that works by finding the a linear function of the data vector that defines a separating hype-plane which separates the data. This happens by minimising the the ratio of within-class variance to between class variance. According to reference [47] the Linear Discriminant function is as follows: "the concept of the discriminanat function y(x) is that the vector x is assigned to class c_1 if $y(x)>0$ and to class $c_2$nif $y(x)<0$"

$$y(x) = w^T x + w_o \qquad (1)$$

"where w is the weight vector and parameter w_o is bias or threshold. This form is optimum of class-conditional densities having normal distributions with equal covariance matrices."

So the function to be generalized

$y=g(w^T+w_o)$ (2)

"where g(.) is called the activation function and consider a two class problem in which the class-condition densities are given by Gaussian distribution with equal covariance matrices $\Sigma_1=\Sigma_2=\Sigma$ so that"

$$p(x|c_k) = \frac{1}{(2\pi)^{d/2}\,|\Sigma|^{1/2}}\, exp\{-1/2(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)\} \quad (3)$$

Using Bayes theorem, the posterior probability of membership of class c_1 is given by

$$P(c_1|x) = \frac{P(x|c_1)\,P(c_1)}{P(x|c_1)\,P(c_1)+P(x|c_2)\,P(c_2)} \quad (4)$$

$$= \frac{1}{1+exp(-a)} \quad (5)$$

$$= g(a) \quad (6)$$

Where

$$a = ln\frac{P(x|c_1)\,P(c_1)}{P(x|c_2)\,P(c_2)} \quad (7)$$

And the function g(a) ia a logistic sigmoid activation function given by

$$g(a) \equiv \frac{1}{1+exp(-a)} \quad (8)$$

If substitute expression for class-conditional densities from (3) into (7)

$$a = w^T + w_o \quad (9)$$

Where

$$w = \Sigma^{-1}(\mu_1-\mu_2) \quad (10)$$

$$w_o = -\tfrac{1}{2}\mu_1^T\Sigma^{-1}\mu_1 + -\tfrac{1}{2}\mu_2^T\Sigma^{-1}\mu_2 + ln\frac{p(c_1)}{P(c_2)} \quad (11)$$

"the use of the logistic sigmoid activation function allows the output of the discriminant to be interpreted as posterior probabilities. This implies that such a discriminant is providing more than simply classification decision , and is potentially a very powerful result"

2.6.3 Cross-Entropy Error for two classes

As this thesis deals with two classes classification problem, so an explanation of cross-entropy error function for two classes would be appropriate. According to [47], "in solving a problem involving two classes a network is considered with single output y. Value of y represents the posterior probability $P(c_1|x)$ for class c_1. The posterior probability of class c_2 will then be given by $P(c_2|x)=1-y$. This can be achieved when t=1

if the input vector belongs to class c_1 and t=0 if it belongs to class c_2. These two expression could be combined into single expression

$$P(t|x) = y^t(1-y)^{1-t} \quad (1)$$

which is a particular case of the binomial distribution called the Bernoulli distribution. With this interpretation of the output unit activation assuming the data points are drawn independently from this distribution, is then given by

$$\prod_n (y^n)^{t^n}(1-y^n)^{1-t^n} \quad (2)$$

It is more convenient to minimize the negative logarithm of the likelihood. This leads to the cross-entropy error function in the form

$$\mathrm{E} = -\sum_n \{t^n ln y^n + (1-t^n) ln\,(1-y^n)\} \quad (3)$$

2.6.4 Regularization Parameter

The regularization parameter is added to the error function to penalize as the model becomes complex [46].

$$\mathrm{E} = \mathrm{E}_{data} + \alpha\, \mathrm{E}_{\mathrm{reg}}$$

Where E_{data} is the data error function and E_{reg} is the penalty that increases as the model becomes more comples and α is a regularization parameter.

2.6.5 Validation and Cross validation

It is important to evaluate the performance of the classification as stated in [47]. To do that one must evaluate the performance, by evaluating the error function for example, using data different than the data used in training process. Therefore, to create a validation data, the training data is divided into training and test data. This technique is called cross-validation method and it is used to avoid over-fitting. A cross-validation is a technique of dividing the training data into training and test set. There are three types of cross-validation, the leave one out cross validation, holdout cross validation and K folds cross validation. The main idea of cross validation as explained in [47] is "the training set is divided at random into S distinct segments. Then train the classification tool using data from S-1 of the segments and test its performance, by evaluating the performance, using the remaining segment. This process is repeated for the S possible choices for the segment which is omitted from the training process." The difference between the kinds of cross validation is in leave one out cross validation is that all data is used for training except one point is used for validation. The holdout cross validation is dividing the training data randomly into two groups. Normally less

than third of the training data is selected for validation set. K folds cross validation is dividing all the training data into equally k-folds. All of the folds are used in the training except one fold for testing.

2.6.6 ROC analysis

As the experiments in the thesis depends on the ROC curve to compare the performance of the classifiers, a definition of the ROC analysis is needed. According to [46] the need for ROC curve appears in classification problems of two classes. So that, in measuring classifying accuracy one needs to calculate the ratio of misclassified inputs. The definition of the ROC analysis is [46] "In classification problems, the task is to allocate new or previously unseen examples x to one of two classes. This is generally based on a model, or set of models, induced from some exciting corpus of data whose true classes are known already. The misclassification rate (proportion of data which is labelled with an incorrect class by the classifier) is taken as a measure of classifier accuracy, and as objective to be minimized. However, when there is an imbalance in the cardinality of each distinct class in a set of data, for training and /or testing, the total misclassification rate can be misleading. In order to deal with class imbalance, Receiver Operating Characteristics Analysis is typically used in the 2-class classifier optimization. This analysis traces out the true positive rate (the proportion of correct assignments to the principal class by the model) against the false positive rate(the proportion of incorrect assignments of the second class to the principal class by the model) by varying the classification threshold of the model (if the model outputs a probability of assignment, or a score) or the parameters of the model itself. This visualization shows the trade-off between the accuracy in classifying the two separate classes for a particular model. The best classifier would operate in the top left of the plot, with a TPR of 1 and FPR of 0.

2.7 Conclusion

To conclude, the field of authorship attribution is full of interesting challenges. Many researchers have tackled this field in order to solve some of the well-known disputed authorship problems. The main challenge or obstacle that faces studies and researchers is finding the best features, which can clearly extract the author's style or fingerprint and so be used by scholars to differentiate between different authors. Some scholars use features such as vocabulary and sentence richness, others try to find

features that are not related to the grammar of the language but relate to the text itself, such as character-level and word-level features. The latter direction of research led to the appearance of function words in this field, which has been very successful, because it has been found that writers and authors use function words unintentionally and in a characteristic way; an individual author rarely controls or changes the way he or she uses function words.

Another factor that comes into play, in the context of finding suitable features, is the issue of feature selection. That is, one can find the right features (e.g. either characters, function words, word sequences), but having decided to use function words (for example), there may be a very large collection of function words, and the problem now is to select which of them should be used. The goal of feature selection is to try to select only the important features that can play a part in reflecting the author's style. For example, if we use function words as features, it is not advisable to use all functions words because their number may be too large. This means that the classification algorithm will be burdened with extra, and maybe excessive, computational demand; also including all of them will likely include many features that do not help in the classification, and in fact will hinder the classification process by introducing noise. There is a large field of research into feature selection. In [13] Stamatatos did a survey on modern methods of author ship attribution. the survey includes section about feature selection and it shows how feature selection is important in reducing dimensionality. Typical approaches vary from using simple statistical correlation, to choose features that correlate well with the target classification, to using search methods like genetic algorithms to search subsets of features.

The second challenge or obstacle is the classification tool. It is very important to find a suitable tool or method to classify disputed texts according to the features extracted from the dataset. Many classification tools have been used in this field. However, the way these tools have been used with different kind of features and on various datasets leads to uncertainty about which tools might be the best and which features might work best across many datasets.

I would argue that the best approach is to find a generally good choice of feature, and a generally good classification tool. A generally good choice of feature is one that works well across different datasets and languages. Finding such a choice of feature would be a significant step in the authorship attribution field. This could be

more easily achieved (and recognised and accepted) if experiments and studies in authorship attribution shared datasets and classification tools much more commonly than they do now. For example, we can compare the accuracy and performance of different types of features if we conducted experiments on the same datasets and using the same classification methods. Then after deciding which type of feature is most promising, we can then conduct experiments using different classification tools; however, it remains the case that different authorship attribution example problems may be quite different in their difficulty. It depends on the amount of difference in style between the different authors involved. For example we can expect that simple classification methods like *k*-nearest neighbour or Naive Bayes might perform very accurately on a simple problem, but more sophisticated tools are needed for cases where the different authors have similar styles. This suggests again that choice of feature is the more interesting and important issue in the authorship attribution field.

However there is another issue concerning the classification tool. In authorship attribution and related studies, there is often also an interest in stylometry. That is, we are interested in solving a specific authorship attribution problem, but we are also interested in understanding what are the very specific collection of features (e.g. if we using function words, which words are most important?) that characterize a particular author, or which are generally important for distinguishing between many authors. For example, there might be 200 function words, but maybe it is possible to distinguish between authors only by focussing on the way they use a small specific subset of these words. It is interesting to know this subset. We mentioned the problem of feature selection above, which is closely related to this issue.

In this thesis the main aim is to find a way to approach the authorship attribution problem for the Arabic language. As a start, I looked at a classification method that was able to explore using small numbers of features, and tested it on famous disputed problems of works in English literature which are available for researchers such as Federalist papers, Pickett Letters and 15th book of Oz. I looked at function words and character level features, both found promising in the review of the literature. After showing that our classification method, linear discriminant analysis hybridised with evolutionary search, performed very well on the classical English cases, I focussed on finding function words in the Arabic language. As we will see, the performance of these function words on Arabic datasets, using the same linear

discriminant hybrid method, was very good. Also we make the Arabic datasets available to all researchers. In this way we make a helpful step towards discovering general language independent features and tools.

3

English Authorship Attribution

Chapter overview

In this chapter I describe experiments in the field of Authorship Attribution for English texts. Section 3.1 provides an introduction to the main research ideas and goals of the work in this chapter. Section 3.2 presents the datasets used in this chapter, and Section 3.3 describes the text features used in these experiments, which is a standard collection of English function words. Section 3.4 discusses our hybrid evolutionary algorithm, and issues such as evaluation and comparison of feature subsets. Section 3.5 presents a number of experiments and their results. Finally, Section 3.6 presents the conclusions of this chapter.

3.1 Introduction

In this chapter I present the work I did in the field of authorship attribution using the English language. This work was partly done before the PhD research, and what is reported in this chapter focuses on the additional work that was done as part of the PhD research. The main research idea is to solve the authorship attribution problem in the English language using English function words as a feature and an evolutionary algorithm to select the best features to classify with. The classification tool used was the Linear Discriminant Analysis which is described in details in chapter 6. The research was done on two stages. The first stage was the main phase which most of the work done in it and it was the thesis in my master degree. The second phase is about introducing a new feature selection and evaluation methodology other than the one used in the main research. Also, a comparison between the two methodologies used in selecting and evaluating the features. This research is published in 2007 IEEE Congress on Evolutionary Computing [27].

The dataset used in the research was the federalist papers and the 15th book of Oz. These two works are considered well known and a good dataset to rely on as mentioned in chapter 2.

The features used in this research were function words. The function words used were the well-know list of 70 words suggested by Mosteller and Wallace [15]. The relative frequencies of the 70 function words were calculated out of the dataset. These frequencies are treated in vector form of 70 variables.

In this research, the basic setup of the experiments is different from that used in the Arabic experiments. In the case of Federalist papers, the basic setup for the experiments depends on the number of distinct texts or articles from each author, not on the number of words 'chunk size' as in the Arabic Authorship Attribution experiments in chapter 5. That is, in this chapter each individual text or article represents one vector. This is due to the small average number of words of the texts.

In the case of The Book of Oz, the basic setup of the experiments depends on the number of chapters, not on the number of texts, as in the federalist papers. Each vector represents two chapters. This is due to the large average number of words of the authors' books. In both cases, the experimental setup used is consistent with that used

by other researchers who have investigated Authorship Attribution for the Federalist Papers and the Book of Oz.

3.2 The Data Set

The materials used in the project were the Federalist papers and the collection of Books of Oz. These are considered the most famous disputed articles and books in the history of English literature.

The data was downloaded and transferred into plain text format. Then, the frequencies of the 70 function words were calculated by a developed software program. The frequency of the function words of each piece of work was treated as one vector. Each vector consists of 70 variables.

The first data set is the federalist papers. These are 85 essays published anonymously. There are three authors of the 85 essays. Alexander Hamilton wrote 51 essays and James Madison wrote 15 essays and John Lay wrote only five essays. Both Hamilton and Madison wrote three essays together. For the remaining 11 papers, both Hamilton and Madison each claimed to be the author.

The average size of the 51 undisputed essays of Hamilton is 2203 words. The smallest essay is 987 words and the largest essay is 5733 words. The average size of the 15 undisputed essays of Madison is 2755 words. The smallest essay is 1898 words and the largest one is 3575 words. The average size of Jay's undisputed essays is 1704 words and the smallest essay is 1366 words. The average size of the disputed papers overall is 2022 words. The smallest essay is 1133 words and the longest one is 3056 words.

The frequency of the 70 function words is calculated for each paper separately. The total number of vectors produced is 77 vectors. Each vector has 70 variables representing the features of every paper. They are divided as follows. 51 vectors are for Hamilton, 15 vectors are for Madison and 11 vectors are for the disputed papers.

The papers were downloaded from the internet from the CNSNEWS.com (http://www.cnsnews.com/Library/Federalist/Default.htm) and the library of congress website (http://thomas.loc.gov/home/histdox/fedpapers.html).

The second data set is the 15th Book of Oz. The 15th Book of Oz was first published one year after the death of Baum under his name. Then Thompson claimed that she is the author of the 15th Book of Oz.

All of the 14 books of Baum and only five books from Thompson found electronically on the internet. The average number of words in the Thompson books is 39017 words. The smallest book is 33842 words and the largest book is 45654 words. The average number of words of the six books of Baum is 42100 words. The smallest book is 38413 words and the largest book is 53206 words.

The relative frequencies of function words were counted in every two chapters because the number of words in each book is very large. Each book contains about 24 to 30 chapters. That is, we divided each book into 12 to 15 parts and calculated the relative frequency of each part. We calculated only seven books of the Baum collection and five books of the Ruth Thompson collection. We ended with a total number of 147 vectors for both writers and the disputed book. Frank Baum has 86 vectors and Ruth Thompson has 49 vectors and 12 vectors for the disputed Book of Oz.

The 14 books written by Lyamn Frank Baum were all downloaded from Literature.org (http://www.literature.org/authors/baum-l-frank). Only five books by Ruth Plumly Thompson were downloaded from "the online book page" (http://onlineBooks.library.upenn.edu).

3.3 The English Function Words

The function words in the English language are words that have no significant meaning but play an important role in link the words together and help in expressing the whole meaning. The function words also have grammatical role in the English language, which is adding grammatical information. They are pronouns, conjunctions, prepositions, auxiliary verbs, and some adverbs. In addition, it is known that function words are hardly or rarely borrow from other languages and they are no new ones could be created. All this reflects that the usage of function words depends on the author style rather than depends on the English grammar. Therefore, the writer will use the function words unintentionally, which will reflect radically the writer's style and pattern. In a great psychological study of function words by C. Chung and J. Pennebaker called *The Psychological Functions of the function words* [33]. They stated in the study that just the 10 most common used function words, which resembles (function words) only 0.04% of any native speaker vocabulary, used as much as 20% of the words we use every day. They added “In daily conversations, however, we have virtually no control or memory over how and when they are used either by speaker or by ourselves. Despite

rarely paying them any conscious attention, function words have powerful impact on the listener/reader and at the same time, reflect a great deal about the speaker/writer."

Also as discussed in chapter 2, researchers as A. Shlomo and L. Shlomo in their paper [32], stated that "due to their high frequency in the language and highly grammatical roles, function words are very unlikely to be subject to conscious control by the author. At the same time, the frequencies of different function words vary greatly across different authors and genres of text, hence the expectation that modelling the interdependence of different function word frequencies with style will result in effective attribution." Also, researcher Foster's studies claim that authors are using certain pattern of words that reflects his style as discussed in detail in chapter 1 and 2. As a result of the success achieved by Mosteller and Wallace [15] by using the relative frequencies of function words in solving Authorship Attribution problems, many researchers now use the relative frequency of function words in their research such as papers [3],[5],[7],[8],[10],[15],[16], [29] and [32].

a	All	Also	an	and
any	Are	As	At	be
been	But	By	Can	do
down	Even	every	for	from
had	Has	Have	her	his
if	In	Into	is	it
its	May	More	must	my
no	Not	Now	of	on
one	Only	Or	our	shall
should	So	Some	such	than
that	The	Their	then	there
thing	This	To	up	upon
was	Were	What	when	which
who	Will	With	would	your

Table 3-1 Mosteller and Wallace Primary function words

Despite of the use of different tools of classification, researchers agreed to use the relative frequency of the function words as features, which reflects the style of the writer. For example, Binongo [10] used function words in his research to solve the problem of the disputed Book of Oz. Fung [3] used function words in his paper and he tried to select the best subset of these function word to help him classify the disputed works. In addition, Bosch and Smith [29] found a separating hyper-plane between authors, using the relative frequencies of function words.

3.4 The Evolutionary Algorithm

The evolutionary algorithm's goal is to find good and small subsets of features that can classify the disputed texts correctly. Initially, the algorithm selects randomly from the 70 function words. The fitness of a subset of function words is tested using the LDA classifier, and quantified according to the area under the ROC curve. If a subset of function words is in this way able to classify all the training data set correctly then we apply these frequencies on the Federalist disputed papers and the disputed Book of Oz. The same algorithm is used in the Arabic experiments and is explained in details in chapter 5.

There are two methods for selecting the best candidate function words. As mentioned in the introduction of this chapter, the feature selection in the first phase of the research is called ROC-Dominance based approach and the second type is the Area Under ROC-fitness approach.

3.4.1 ROC-Dominance based Approach.

Finding a good ROC curve is the most important issue in selecting the best set of features to classifying the disputed work. The more the test data is classified correctly, the larger the area under the ROC curve is achieved. The largest area under ROC curve is the result of classifying all the training data set correctly, which generates a true positive ratio of 1, and a false positive ratio of 0.

In the ROC-Dominance based approach, we use Pareto dominance, as in paper [30], instead of area under the curve, as the way to select the best ROC curves. That is, each time a new ROC curve is produced, each classifier of the new ROC curve is compared with the best ROC curve in the current population. A new classifier dominates the old classifier if the true positive ratio value of the new classifier is larger than the true positive ratio of the old classifier and the new classifier's false positive ratio is smaller than the false positive ratio of the old classifier.

The initial population contains only randomly chosen singleton feature vectors. That is, the run starts with only one candidate and only one ROC curve produced as a result of evaluating this candidate. Then another new ROC curve is produced which in turn is compared to the old one or (the best ROC curve of the past runs). Then according to this comparison a new ROC curve is produced out of the two curves by selecting the dominants classifiers out of the two curves (the new one and the best one).

This only happens if the new ROC curve has some classifiers better than the best ROC curve. But if the classifiers of the newly produced ROC curve are all dominated by the best ROC classifiers, then there is no point of producing a new curve as the best ROC curve is still better. Finally, the best ROC curve is a combination of best classifiers out of all the ROC curves produced during all runs.

3.4.2 Area Under ROC-Fitness Approach

Our second approach also wraps a simple EA around the LDA training process, but this time simply calculates the area under the ROC curve, and treats this as a single-objective fitness value to be maximized. For this approach we use a straightforward small-population steady-state evolutionary algorithm. Specifically, population size 5, binary tournament selection, and replace-worst replacement, breaking ties by number of features. That is, in each generation, binary tournament selection is used to choose a parent. A mutant is then generated and evaluated. The mutant enters the population if it is at least as fit as the current worst. If there is a tie between the mutant fitness and the fitness of the current worst, but the mutant contains more features than the current worst, then the mutant is discarded.

3.5 Experiments and Results

3.5.1 Federalist papers principal component analysis

Following standard PCA applied to the Federalist Papers data, we plotted the projections of the data onto the first two principal components in Figure 3-1. It can be seen that the general positions of the eleven disputed papers (squares) are not able to be clearly distinguished from either the Madison papers (crosses) or the Hamilton papers (circles). Arguably, they are shifted more towards the ‘Madison space’ than the ‘Hamilton space’, but several individuals are much closer to Hamilton papers than to any Madison paper.

Following PCA, we used the principal component transformations of the data as input to the LDA process already described, trying this for the first *k* principal components, for each *k* from 2 to 70. For each such *k*, we measured validation error by recording the cross-entropy error on the disputed papers. Figure3- 2 shows the plot of validation error against number of principal components used. Clearly, the findings from PCA are that we need the first 15 principal components to reach the minimum error, tentatively

suggesting that around that many function words may be needed (in the sense that this is the suggested number of latent features required for good performance).

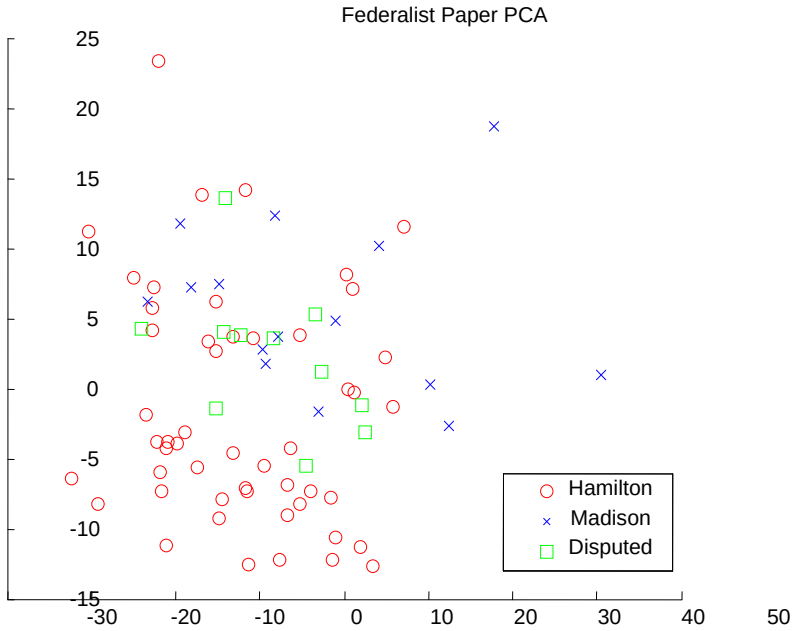


Fig 3-1. Projection of the first two principal components of the Federalist Papers data.

We tried to find a separating plane using the principal components, and to find a least validation error generated by the LDA using the 70 principal components.

In Fig. 3-2 We found that the first 15 principal components has the lowest error generated. This indicates that it needs 15 principal components to reach the minimum error, which is not small number, while according to our algorithm (as we will see in later results) we need only two features to separate the two classes.

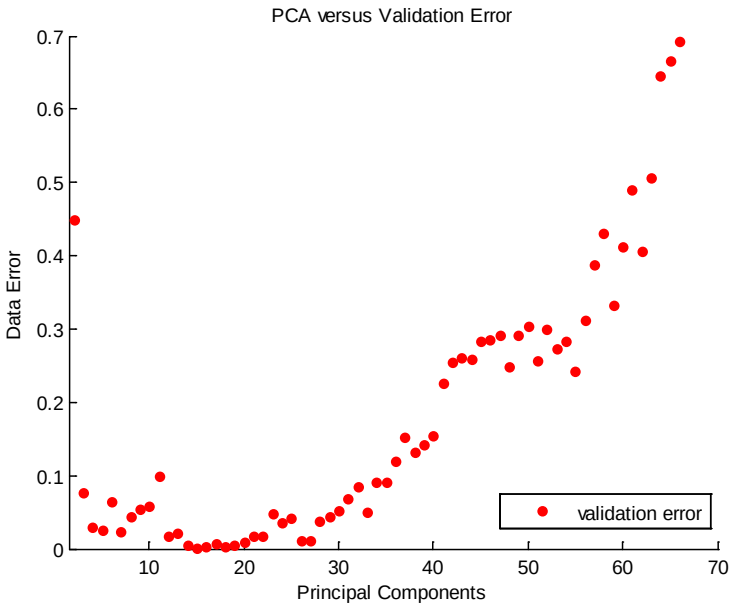


Fig3-2. Error on the disputed papers against number of principal components learned from the test data: Federalist papers.

Fig. 3-3 presents the training error and the validation error graph of the first 15 principal components. We trained the LDA to find a separation hyper-plane for the output of the PCA and it indicates that the validation error of the first 15 principal components is the lower error obtained.

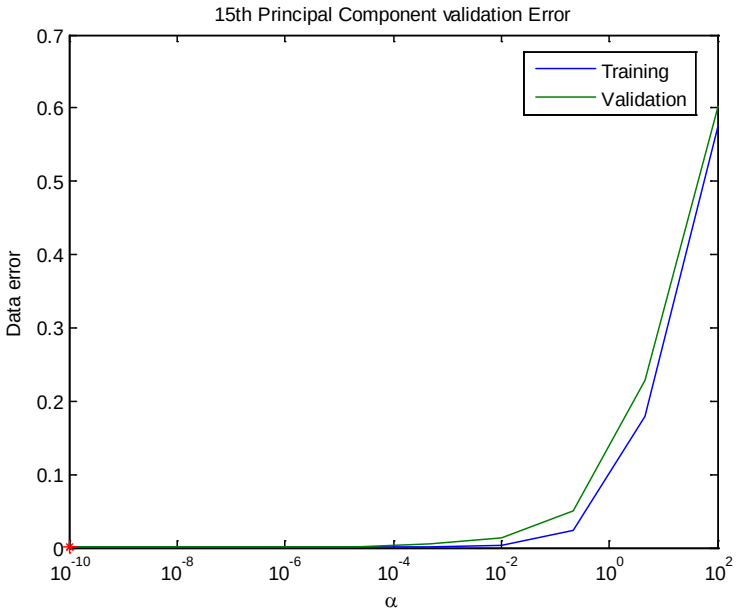


Fig. 3-3 The first 15 Principal Component Validation Error

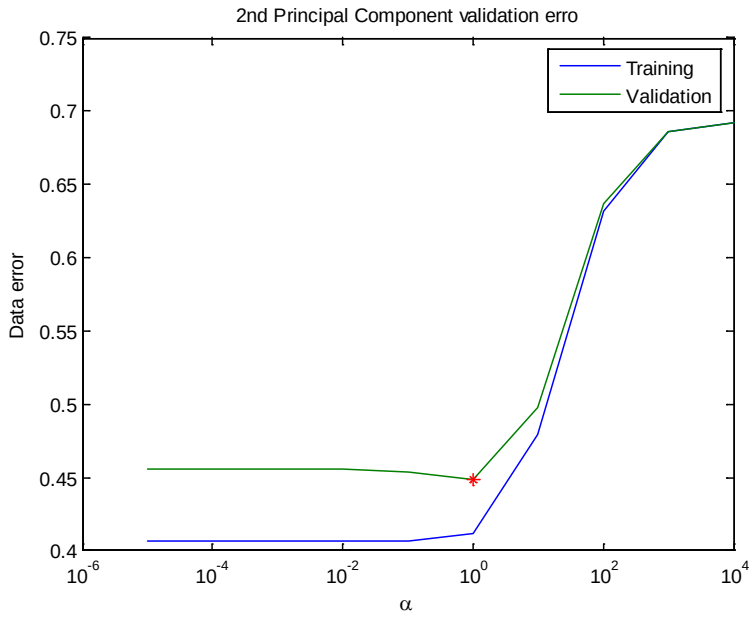


Fig. 3-4 The first two Principal Component Validation

Also, we plot training error and validation error of the first two principal components to show that a small number of principal components cannot classify correctly. Fig. 3-4 presents the training error and the validation error graph of the first two principal components.

3.5.2 Federalist Papers: Evolutionary algorithm using ROC-Dominance Method

In the evolutionary algorithm program, we depend on the output of the ROC curves, which reflects the performance of the relative frequency of the function words in classifying the training dataset. That is, in the Federalist paper case, Fig. 3-5 represents one of the ROC curve outputs after running the evolutionary algorithm for 50 iterations. The algorithm tries to compare the classifiers and create the best ROC curve or the best classifiers by selecting the dominating classifiers and adding the non-dominating classifiers into the population. In Fig. 3-5 it can be seen that the algorithm selects the best two classifiers (red circles) that dominates all other classifiers in the graph. The green lines are all the ROC curves obtained during the 50 runs. The red curve is the best ROC curve achieved in this particular graph. It is clear that the best curve is the one closer to the left corner, maximizing true positives and minimizing false positives ratio.

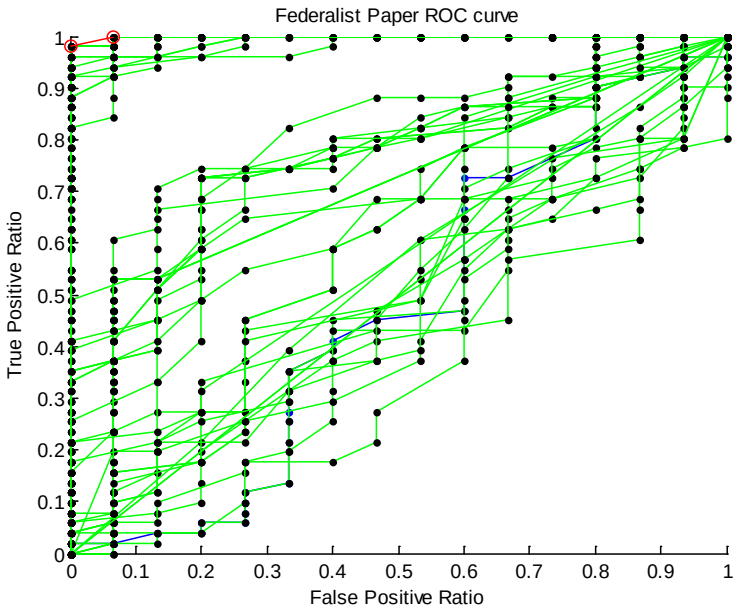


Fig. 3-5 50 runs of Federalist papers ROC curve

Fig. 3-6 shows an optimal ROC curve reached after 65 iterations. In this case, the algorithm reached this best ROC curve by finding the function words which classified all the training data set correctly. This performance reflected on the graph is 100 % accuracy. This curve was obtained with three different feature sets, each of which classified all the training data set correctly.

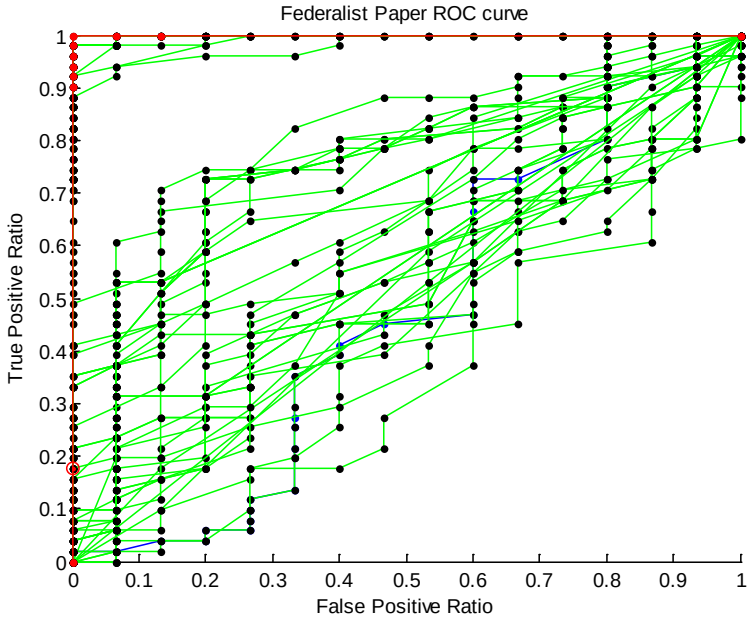


Fig. 3-6 Federalist papers Optimal ROC curve

These sets of function words are: **(some, upon, had, into**) ,(**thing, that, upon**) and finally, using only two function words: (**upon, with**).

3.5.3 Book Of Oz: Principle Component Analysis

Principal component analysis applied to Oz data does not clearly help in classifying the disputed books. First, we tried to plot the principal components using every book as one data vector. That is, only five vectors for Baum books and only five vectors for Ruth Thompson's books. Fig. 3-7 represents the principal component analysis for the Oz collection. As in the Fig. 3-7, it is hard to classify the disputed books accurately because it is not in the centre of the Ruth space or the Baum space. Also, there is one book, which belongs to Baum (red circles) that lies very near to the Thompson space. Therefore, one cannot classify the disputed book with confidence using PCA.

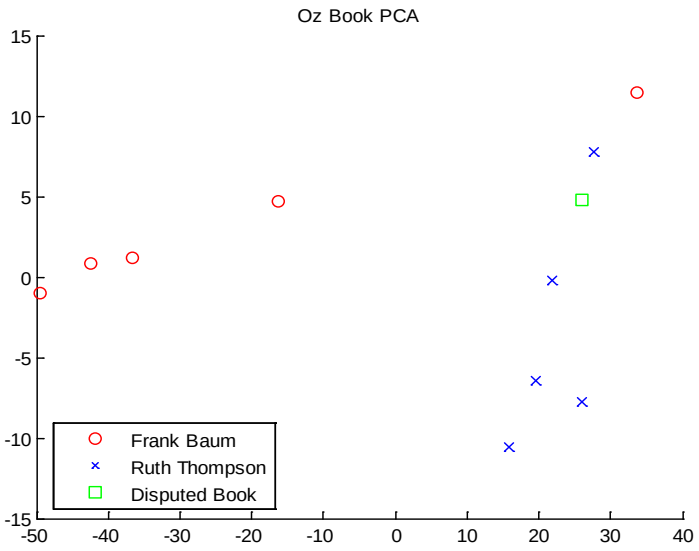


Fig. 3-7 Oz Books Principal Component Analysis

Second, we plotted the principal components of the Oz books using a different data set in Fig 3-8. In this case each book is divided into several vectors. That is, we counted the relative frequencies of the function words for every two chapters of every book. Seven books for Frank Baum and five books for Ruth Thomson are plotted in Fig. 3-8. As in Fig. 3-7, the Ruth Thompson data space (blue x's) and the Frank Baum data space (red circles) are not separated totally. That is, there is an overlap in some points in-between the two spaces. Also, the vectors of the disputed books lie in the Ruth Thompson space but one cannot classify all of the disputed vectors confidently because some of the Frank Baum vectors lie in the Ruth Thompson space and two disputed vectors lie in the Frank Baum Space, and in fact all of the disputed books are close to

the area between the two spaces. So it is very hard to classify the disputed books with confidence using PCA.

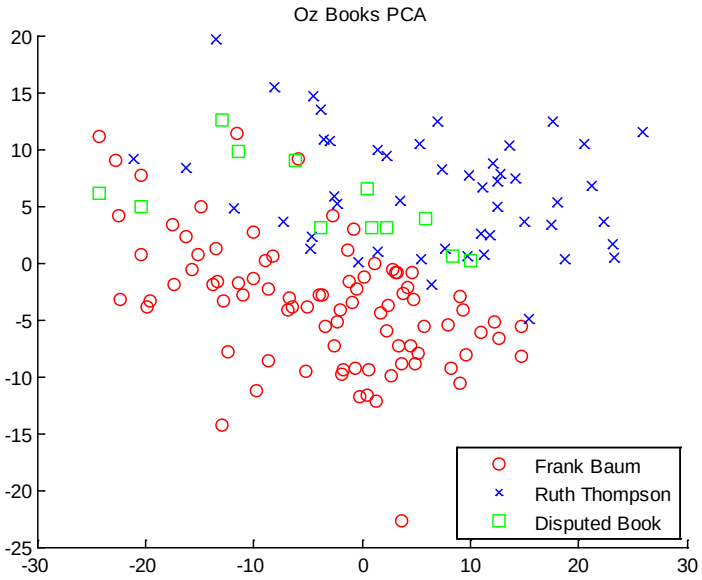


Fig. 3-8 The PCA projection of Oz Books divided in several vectors

Fig. 3-9 presents the validation error generated versus the number of principal components used for the Book of Oz data. We found that the first 26 principal components have the lowest error generated, which is nearly zero. This indicates that it needs the first 26 principal components to reach the minimum error, which is not a small number, while according to our algorithm we need only 14 features to separate the two classes.

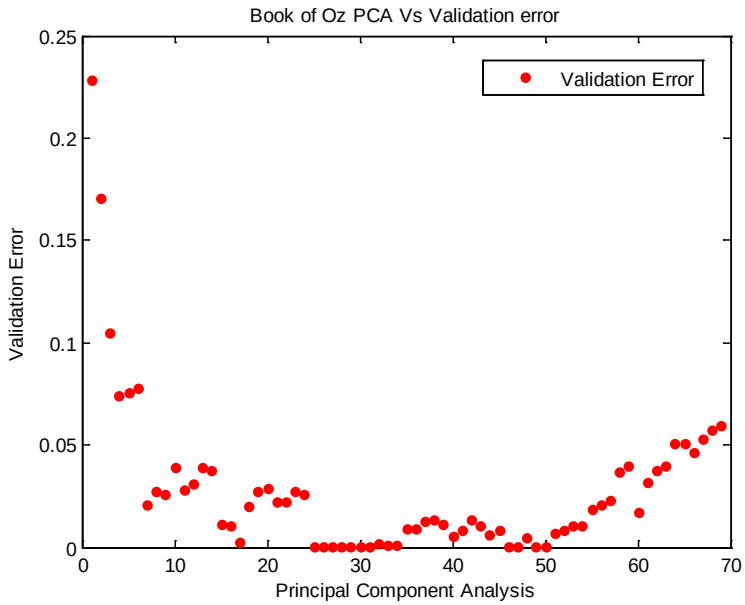


Fig.3-9 Book of Oz PCA versus Validation 2

That is, the case of the Book of Oz is seemingly more complicated than the case of the Federalist Papers. It requires more features and PCA components to classify correctly; this could be because the styles of the different authors in the Book of Oz case are more similar to each other than in the Federalist Papers case.

Fig. 3-10 presents the training error and the validation error graph of the first 26 principal components. We trained the LDA to find a separating hyper-plane for the output of the PCA and it indicates that the validation error of the first 26 principal components is the lower error obtained.

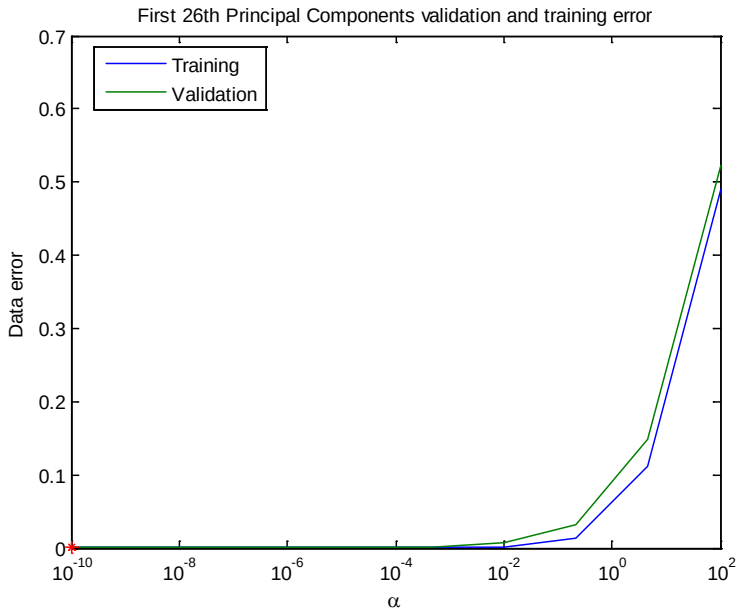


Fig. 3-10 First 26th principal components validation and training error of Book of Oz

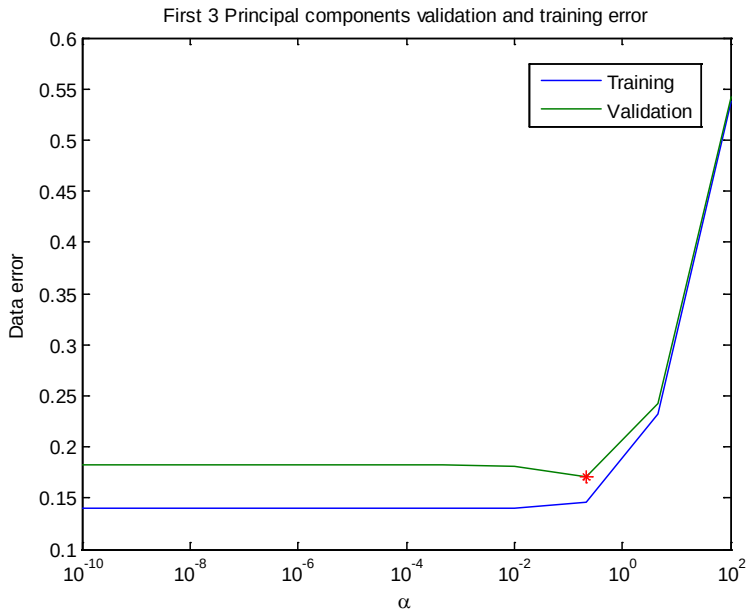


Fig. 3-11 first 3 Principal component Validation and training error of Book of Oz.

It is show that in fig 3-11 the validation error in the first three principal components is much higher than that of the first 26th principal component in Fig.3-10. It shows also, that a small number of principal components cannot classify correctly in this case.

3.5.4 Book of Oz: Evolutionary Algorithm using ROC-Dominance Method

In this section we present some of the results of the evolutionary algorithm program results using the ROC-dominance method, on the Book of Oz data. Fig. 3-12 shows a very small number of iterations. There are only 5 iterations (generations, in the sense of the evolutionary algorithm), which shows how the ROC curves improve with each iteration in the early stages. That is, the blue curve is the first ROC curve obtained by the evolutionary algorithm. This blue curve is the result of the first iteration, which usually uses only one feature. Then, the evolutionary algorithm starts to create new features by either adding, deleting or replacing one of the function words as discussed in detail in chapter 5. Performance is tested in each run by comparing the new ROC curve to the best one previously obtained until it reaches the best ROC curve, which is the red curve in Fig. 3-12. The red curve or the best ROC curve is a combination of all best classifiers obtained from many curves. That is, the best ROC curve is a collection of

best performances of different feature sets. Also, it is clear that the red ROC curve has the greater area under curve in the Fig. 3-12.

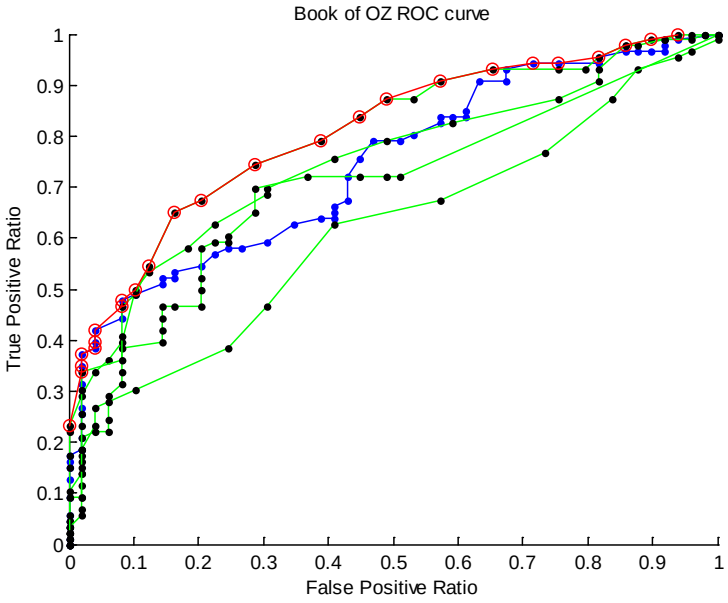


Fig. 3-12 ROC curve for the Book of Oz

3.5.5 Federalist papers: ROC-Dominance vs Area under ROC

We ran each of the Hybrid ROC-dominance approach (ROCD) and the Hybrid AUC-fitness approach (AUCF) 10 times with the following parameters We performed 300-iteration runs, where each evaluation incorporated 5 runs of the LDA training process for different values of α randomly chosen between 0.1 and 1. AUCF used a population size of 5. Results are summarized in table 3-2

	ROC-Dominance	Area under ROC
Best of 10 trials after 100 cycles	4 function words, achieving perfect discrimination.	**2** function words, achieving perfect discrimination
Mean of 10 trials after 100 cycles	4 function words, achieving perfect discrimination.	**2** function words, achieving perfect discrimination
Best of 10 trials after 300 cycles	**2** function words, achieving perfect discrimination	**2** function words, achieving perfect discrimination
Mean of 10 trials after 300 cycles	Perfect discrimination always achieved, with mean of 3.1 function words	**2** function words, achieving perfect discrimination

Table 3-2 Roc-Dominance and Area under ROC methods results on Federalist Papers

There is insufficient evidence so far to support a statistical claim that Area under ROC is a better approach than ROC-Dominance on this problem, however the important and interesting findings are that these methods can both reliably obtain classifiers that use only 2 function words, which perform perfectly in assigning authorship of the disputed papers.

In comparison, Fung [3], using support vector machines, found a classifier that used 3 function words (*to*, *upon*, and *would*), while Bosch and Smith [29] achieved the same result with an extensive test that searched all combinations of 1, 2 and 3 function words using a linear programming formulation, discovering only a single set of 3 (*as*, *our* and *upon*) that achieved perfect classification. In our case, ROC-Dominance was able to find a classifier that worked only with *upon* and *with*, while different perfectly-classifying sets of words were found by AUCF, but usually including the word *upon*.

3.5.6 The Book of Oz: ROC-Dominance vs Area Under ROC

We ran each of the Hybrid ROC-dominance approach of section and the Hybrid AUC-fitness approach 10 times with the same parameters. Results are summarized in table 3-3. Again, there is insufficient evidence so far to support a statistical claim that Area unde ROC is a better approach than ROC-Dominance on this problem, however we again have interesting findings that show that each method is adept at reliably discovering relatively small subsets of features that can perform perfected discrimination of the disputed work. Previous work on this case is less common than in the case of the Federalist papers, and we do not have comparable results concerning attempts to minimise the number of features. Binongo [10], concentrated on using principal components of a set of 50 function words. We feel it is therefore an interesting contribution that we have found sets of five function words (e.g. *any*, *are*, *but*, *be* and *my* – most discriminating sets of five words included *be* and *my*) that can lead to perfect discrimination on the (unseen) disputed work.

	ROC-Dominance	Area under ROC
Best of 10 trials after 100 cycles	6 function words, achieving perfect discrimination.	**5** function words, achieving perfect discrimination
Mean of 10 trials after 100 cycles	6.5 function words, achieving perfect discrimination.	5.8 function words, achieving perfect discrimination
Best of 10 trials after 300 cycles	6 function words, achieving perfect discrimination	**5** function words, achieving perfect discrimination
Mean of 10 trials after 300 cycles	Perfect discrimination always achieved, with mean of 6 function words	**5** function words, achieving perfect discrimination

Table 3-3 Roc-Dominance and Area under ROC results for Book of Oz

3.6 Conclusion

In conclusion, we addressed the problem of Authorship Attribution using hybrids of simple evolutionary search and a linear discriminant classifier, using evolutionary search to find small function word subsets as the sets of features used in training the classifier. Baseline comparison was also done with using straightforward PCA to transform the data, finding that 15 and 26 components were needed respectively to obtain perfect performance on the disputed *Federalist Papers* and *Book of* Oz respectively. Using a simple EA to search feature subsets based on iteratively selecting randomly from subsets with so-far nondominated ROC curves (the ROC-Dominance approach), we were able to reliably find subsets of function words of sizes 2 and 6 respectively. Regarding the Federalist Papers task, this is a smaller set than has been achieved before in the literature, which in turn has some implications and interest for stylometry studies. In the case of the *Book of Oz*, we can only conclude that the result seems very good, and perhaps sets a target for related studies, since work so far has not used the *Book of Oz* task in an explicit attempt to minimise the number of function words used for discrimination. A simple EA for evolving ROC curves, again hybridised with the LDA classifier (which we called Area under ROC), achieved slightly better results here than the ROC-Dominance approach, finding sets of 5 function words that discriminated the validation set perfectly in the *Book of Oz* task.

Our work on this so far has been hampered by the long training times required by the LDA classifier built into our fitness function, and the corresponding repeated

runs of that process that are required to find a good parameter for the weight decay regularisation. In ongoing work we will compare this with less time-consuming classifiers, and so it has yet to be seen whether similar or better results can be achieved with a less sophisticated classifier. However, given the results, and comparative work using SVMs and linear programming that has not matched our results on the Federalist Papers in terms of number of function words, it seems reasonable to expect that the cost incurred in training the classifier is warranted – that is, the steps taken to promote generalisation performance are reflected in the ability to find perfect classifiers with small feature subsets. Finally, it is clear that evolutionary algorithms have a potential role to play in the area of authorship attribution, and stylometry in general, particularly in terms of feature selection.

4

Processing Arabic Electronic Text

Chapter overview

In this chapter we discuss and present some preliminary issues related to the experiments that we describe in chapters 5 and 6. Section 4.1 explains the process of finding suitable Arabic texts for our experiments in electronic form. Section 4.2 describes the issues we have faced in processing the Arabic Text documents. Section 4.3 gives details of the datasets (and the associated authored books) used in the experiments. Section 4.4 is a discussion of the differences between the Arabic language and the English language, in which consider their relative positions on the wider ‘family tree’ of languages.

4.1 Finding Arabic Electronic Text

Initial searches showed that Arabic Electronic Texts are very difficult to find on the internet or in any other medium. This problem occurs because there is not much Arabic literature transferred to electronic form as yet. Most of the Arabic electronic texts that can be discovered are recently published ones. Also, searching for an authorship attribution case in Arabic yielded no results; we can expect that such cases exist, but are very unlikely to be found with the texts in question already existing in electronic form. Searching for Arabic materials in libraries in the UK is also very difficult because of the fact that the main language is English.

After initial experiences, the search criteria were changed to find Arabic electronic texts subject to certain constraints. First, the requirement was for large texts (with a large number of words) such as novels and plays. Second, it was necessary to search for several such texts written by different authors, with at least two or more electronic books or stories or novels per author. This was necessary to provide the chance to compare effectively between authors, while large texts generally make it easier to extract features. Also, there was a requirement to find different authors who worked within similar genres and topics, since this would ensure that the techniques under investigation would focus on the differences between authors, rather than simply differences related to topic.

Such a search almost invariably finds too few books that are available electronically, however finally a web site of the Arab writers Union was discovered. This website very helpfully provides a wide selection of Arabic literature. It has a variety of stories, novels, poetry, plays, children literature and also Arabic literature studies. All of these works are electronically treated and ready to download. A number of authors and books were selected according to the criteria above, and we settled on a collection of six authors and suitable choices of texts for each. The texts chosen are discussed later in section 4.4. Although these were not cases of disputed authorship, it is straightforward to ‘pretend’ that this is the case for the purposes of developing techniques that are able to address such cases. Anyway, this dataset serves well to consider the general and flexible question of whether we can correctly classify a section of text as being written by one author or the other.

4.2 Processing Arabic Electronic Text

Arabic language text in electronic form is encoded as Unicode characters, unlike the standard ANSII character representation for English. Moreover, the Arabic language is rich in 'diacritics', which are certain symbols added to characters which change their pronunciation and meaning. An Arabic word may, for example, have three letters, but the Unicode representation may contain five or more characters because each diacritic has its own Unicode representation.

Software was therefore developed (in java) to work with the Unicode representation and remove unwanted characters from Arabic texts, leaving only words and/or characters to form the basis of word-based and character-based features in the experiments described later. Figure4-1 shows an example of text before processing, containing question marks, periods, exclamation marks, dashes, commas and unwanted whitespace. Figure 4-2 shows the same text after processing, and eliminating unwanted characters.

إسرار؟ أم إعلان-
ساحة في والغاوي، الضاوي قنديله إلى تأوي الليل، فراشات متصيّداً خيمته، إلى يأوي وهو المجذوب، الشيخ دمدم
عنزته إليه وجذب متهالكة، مهدمة وأسوار شامخة وقناطر حجرية، مئذنة سوى منه يبق لم الذي الجامع الواسعة، العتيق الجامع
أزقّة أحد في عابر درويش لـه أطعمها التي الحلوى، لقمة أضراسه تحت ودارت العريض، وجهه فأشرق ضرّتها، تلمس الأثيرة،
:مردّداً القديمة، حلب
!ملوك؟؟ أم مماليك-
.الحقيقة عن البحث وحمى الدوار فأصابه المؤجلة، الأسئلة ووسواس الحيرة إلى وأسلمه كيانه، فقلب
!بلوى؟ أم حلوى-
حلب، ونسوان حلب، وأسواق حلب، فذكر فرّت، قطاة وبيوض أفعى بين محاصر قنفذ مثل أشواكه يشهر وهو أعلن
في المساجد لزم والتحوّل، للخفاء فرصة أعطته فأغرته، وقاتلاً، مفرداً، جاءها لقد والطرب، والجنون اللذة مدينة حلب، في والسماع
والتكايا، الزوايا إلى فانتقل يقظاً، كان إن ويحاصره نام، إن يطارده كان فالدم ينام، يكن لم الأرق، بحجارة الكوابيس فرجمته البداية،
يقتله أن فكاد ظهره، على الجلد من قربة يحمل السقاية، في عمل ذلك، كل فترك والخدمة، الطعام من أكثر على يتحصّل فلم
.أخرى مرة الدرويش له ظهر وهناك ليقتات، منه السمك يصطاد قويق، نهر عند رابط حينها العطش،
.الطريقة أهل إلى مسدود الطريق-
:وسأل جأشه تمالك ثم فذعر، لها، قال
!مولاي؟ يا إذن أين إلى-
.العيَّارين درجة من والبداية ..البداية إلى-
..أفهم لا-
.لمسالكه المجرّب العارف وأنت القاع، من تبدأ-
!تعني؟ ماذا-

Fig4-1: Arabic Text before processing and eliminating unwanted characters and space

الضاوي قنديله إلى تأوي الليل فراشات متصيّداً خيمته إلى يأوي وهو المجذوب الشيخ دمدم إسرار أم إعلان مهدمة وأسوار شامخة وقناطر حجرية مئذنة سوى منه يبق لم الذي الجامع الواسعة العتيق الجامع ساحة في والغاوي التي الحلوى لقمة أضراسه تحت ودارت العريض وجهه فأشرق ضرَّتها تلمس الأثيرة عنزته إليه وجذب متهالكة ووسواس الحيرة إلى وأسلمه كيانه فقلب ملوك أم مماليك مردّداً القديمة حلب أزقّة أحد في عابر درويش لـه أطعمها بين محاصر قنفذ مثل أشواكه يشهر وهو أعلن بلوى أم حلوى الحقيقة عن البحث وحمى الدوار فأصابه المؤجلة الأسئلة لقد والطرب والجنون اللذة مدينة حلب في والسماع حلب ونسوان حلب وأسواق حلب فذكر فرّت قطاة وبيوض أفعى لم الأرق بحجارة الكوابيس فرجمته البداية في المساجد لزم والتحوّل للخفاء فرصة أعطته فأغرته وقاتلاً مفرداً جاءها الطعام من أكثر على يتحصَّل فلم والتكايا الزوايا إلى فانتقل يقظاً كان إن ويحاصره نام إن يطارده كان فالدم ينام يكن نهر عند رابط حينها العطش يقتله أن فكاد ظهره على الجلد من قربة يحمل السقاية في عمل ذلك كل فترك والخدمة ثم فذعر لها قال الطريقة أهل إلى مسدود الطريق أخرى مرة الدرويش له ظهر وهناك ليقتات منه السمك يصطاد قويق العارف وأنت القاع من تبدأ أفهم لا العيَّارين درجة من والبداية البداية إلى مولاي يا إذن أين إلى وسأل جأشه تمالك السؤال من تكثر فلا آثم وفمك ملطختان يداك مولاى الحمقى تحتمل لا الحقيقة تعنى ماذا لمسالكه المجرّب

Fig4-2: Arabic Text after eliminating unwanted characters and space.

Having eliminated unwanted characters, the common processing tasks required in our later experiments were to divide a larger piece of text into ‘chunks’ (e.g. of 1,000 words per chunk) and to find the frequencies of certain words or characters within each chunk. Figure 4-3 shows an example of word frequencies in one of the texts. In this case the text was divided into chunks of 2,000 words, and each line in Figure 4-3 represents the frequencies occurring in a separate chunk.

قد 2: كان 7: ثم 5: ذلك 2: ومن 0: فمن 48: من 3: بين 4: كما 54: في 2: حتى 7: ما 37: لا 48: من 18: على 12: أن 3: هو 0: به 4: هذه 1: ألم 2:

5: كان 14: ثم 5: ذلك 3: ومن 1: فمن 49: من 5: بين 3: كما 43: في 4: حتى 13: ما 13: لا 49: من 21: على 11: أن 2: هو 4: به 4: هذه 0: ألم 3: قد

9: كان 14: ثم 5: ذلك 3: ومن 0: فمن 42: من 2: بين 5: كما 41: في 2: حتى 13: ما 16: لا 42: من 17: على 8: أن 1: هو 4: به 4: هذه 0: ألم 1: قد

2: كان 13: ثم 4: ذلك 2: ومن 0: فمن 56: من 6: بين 3: كما 42: في 3: حتى 12: ما 16: لا 56: من 23: على 15: أن 0: هو 2: به 5: هذه 0: ألم 0: قد

قد 6: كان 12: ثم 3: ذلك 0: ومن 0: فمن 51: من 1: بين 3: كما 41: في 3: حتى 9: ما 12: لا 51: من 21: على 8: أن 0: هو 1: به 1: هذه 0: ألم 0:

1: كان 19: ثم 6: ذلك 2: ومن 2: فمن 45: من 5: بين 0: كما 47: في 4: حتى 9: ما 14: لا 45: من 16: على 14: أن 1: هو 2: به 4: هذه 0: ألم 2: قد

كان 13: ثم 11: ذلك 0: ومن 0: فمن 50: من 1: بين 1: كما 53: في 5: حتى 10: ما 19: لا 50: من 25: على 10: أن 1: هو 6: به 2: هذه 0: ألم 2: قد 8:

0: قد 2: كان 10: ثم 4: ذلك 0: ومن 1: فمن 12: من 1: بين 0: كما 13: في 2: حتى 1: ما 3: لا 12: من 7: على 1: أن

Fig4-3: the software output shows the frequencies of certain Arabic words in one of the texts. Each line is the frequencies that occurred in a 2,000 word chunk

In Figure 4-4 we see a screenshot that shows the overall structure of the Arabic text processing software built for the research described in this thesis.

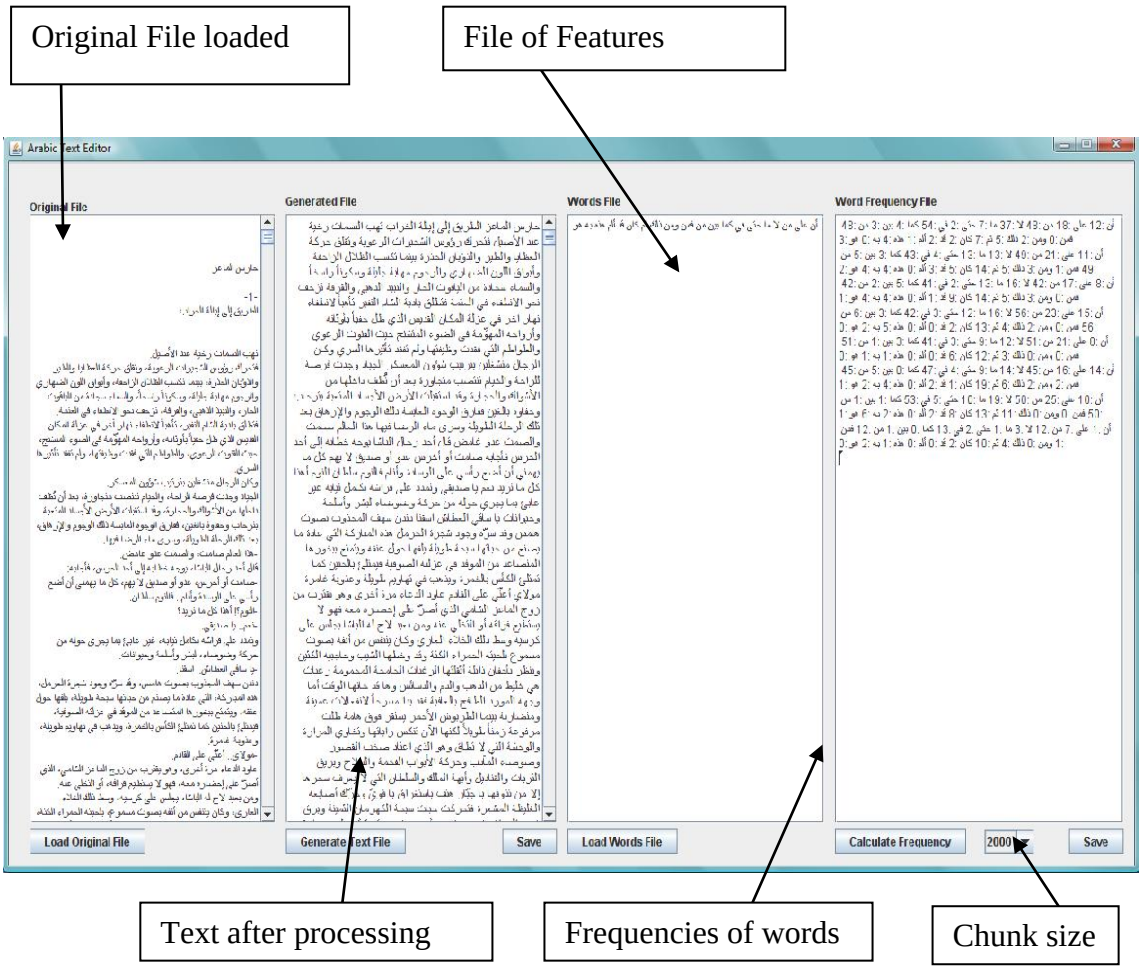


Fig4-4: A screenshot of the software built for processing Arabic text in this research.

The software built and illustrated in Figure 4-4 served for the experiments we describe later that use word-level features. We also did experiments on character-level features, and this required a further application, illustrated in Figure 4-5. This additional application enabled us to generate all of the possible character *n*-grams for some given *n*, to be used as the basis of character-level feature vectors. In Figure 4-5, we see part of the complete set of 784 2-grams that can be created from pairs of the 28 characters in the Arabic alphabet.

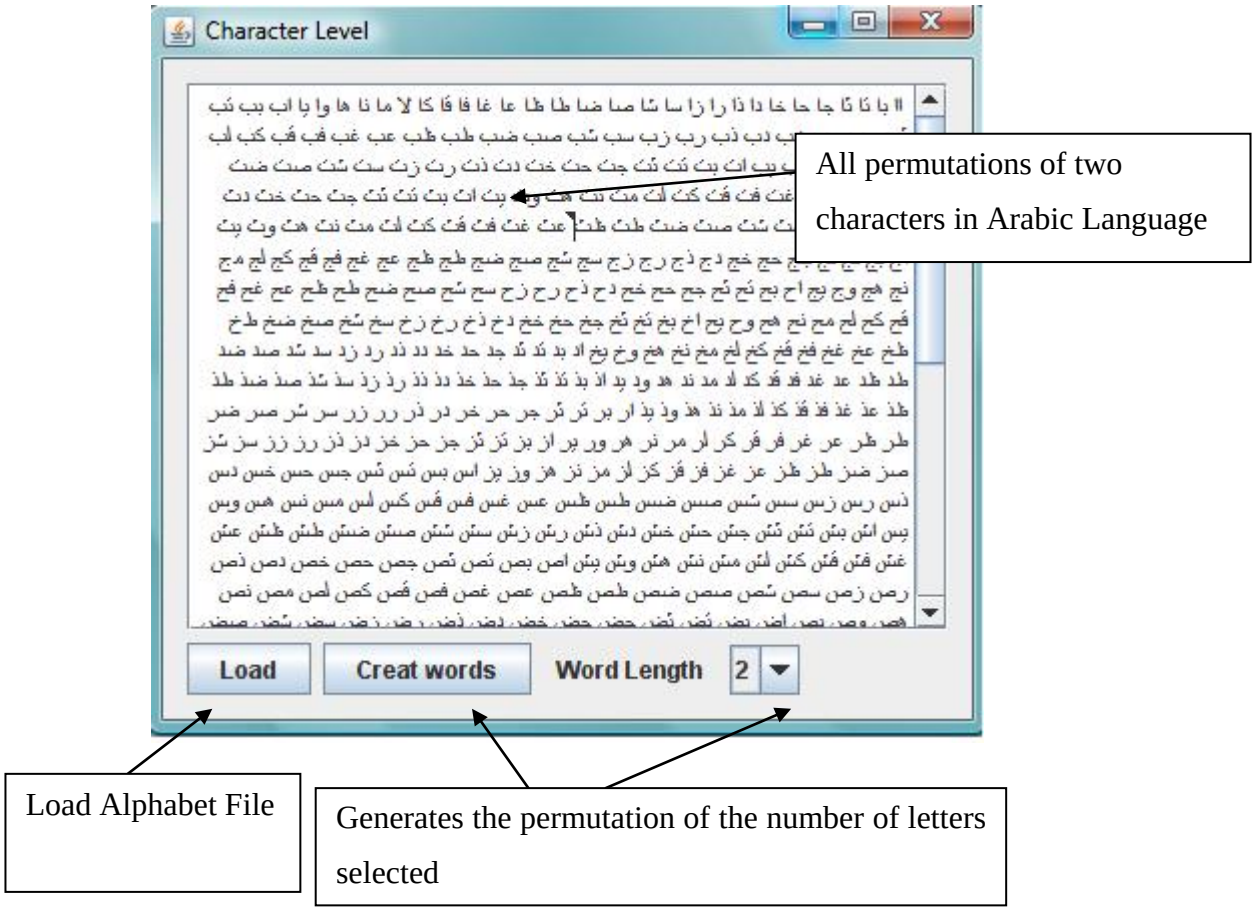


Fig4-5: A screenshot and brief explanation of the character-level software.

4.3 Datasets

The complete dataset consists of 14 books by 6 different writers. As mentioned in the introduction of this chapter, the dataset must be of large texts, each author must have at least two or more texts and finally find authors who write within similar genres and topics. This criterion of searching for books was found in these 6 writers and authors. This criterion was necessary to provide the chance to do experiments effectively between authors. The smallest book contains 13,987 words and the largest one contains 37,567 words. The mean number of words is 23,942. Some of the books are divided into two parts because of their large size. All the books obtained from a website called The Arab Writers Union (awu-dam.net). All the books were downloaded from the web and then all the texts were processed to eliminate the unwanted characters and unneeded spaces. The final material contains only Arabic words separated by single spaces, and no other characters.

4.3.1 Books Summary

The following lists the six authors in turn, and for each author there is a description of the books by that author that are in our dataset.

1- **Ibrahim Khalil**
   a. Haris Al Maiz (Goats safeguard) 14,679 words
   b. Sodom Sebake Al Awez (Sodom the Wild Duck Race) Part1 28,156 words and Part2 29,518 words

2- **Basem Ibrahim Abdo**
   a. Gesr Al Mawt (The Bridge of Death) 37,567 words
   b. Zahra fi Al Remal (Flower in Sand) Part1 23,241 words and Part2 26,061 words

3- **Taleb Omran**
   a. Ahzan Al Sindbad (Sadness of Sinbad) 20,389 words
   b. AlBood Al Khamis (Fifth Dimension) 13,987 words
   c. Madina Kharig Al Zaman (City out of time) 14,513 words
   d. Al Fetiah Al Aghrar we Asfar al Kashf. 19,063 words

4- **Mary Show**
   a. Defly. 24,062 words
   b. Awel Hob and Akheir Hob (First Love and Last Love) Part1 18,848 words and Part2 19,807 words

5- **Mohamed Youssef Salibi**
   a. Al Taih (The Lost) 36892
   b. Sebahaa fi Al Wahl (Swimming in the Mud). Part1 25,647 words and Part2 28,274 words

6- **Hessen Abd Al Kareem**
   a. Al Nabaa. (The source) part1 29,644 words and part2 29,472 words
   b. Shagaret al Toot. (Berries tree) part1 19,024 words and part2 19,998 words

4.4 Differences Between Arabic and English Language

It is estimated that there are around 7,000 spoken languages around the world [19]. According to some authoritative sources, there are 6,909 living human languages [20]. Each language descends from a certain tree or family of languages. There are 16 major families of languages in the world. Each family of languages contains many languages spread over the branches of the family tree, sometimes hundred of languages in one family. Languages of the same family may have a genetic or genealogical relationship between the peoples involved, which leads to linguistic relationships that can be observed [21]. Moreover, languages in the same branch or group within a language family share and have common features such as vocabulary and grammatical features. For example, within the Germanic languages branch, some languages share a high percentage of similarity in vocabulary [23]. The German language and the Dutch language have around 75% similarity in vocabulary, while the similarities between German and Swedish and Danish and Norwegian are around 60% in vocabulary. The French language, which belongs to the Romance branch, shares 85% of its vocabulary with the Italian Language, and around 80% with both Spanish and Portuguese.

On the other hand, there are some languages which form one tree, such as Japanese and Korean, having no genealogical relationship with other languages. Most of the languages of such nature are distinct. The most spoken language in the world is Mandarin, which comes from the Sino-Tibetan Languages Family. Seven out of the ten most spoken languages in the world, including English, come from the Indo-European languages family. Arabic is the fifth most spoken language, and comes from the Afro-Asiatic languages family.

Arabic and English are two quite different languages. They are different in their origins and they come from two different families of languages. Not surprisingly, they are different in structure and nature. The origin of these two languages is totally different. Figure 4-6 shows the origin of the English Language. The English language is descended from the Indo-European Family [24], one of the largest language families, containing most of the major languages of Europe, the Iranian Plateau and South Asia. It contains many of the most spoken language in the world such as Spanish, English, Portuguese, Russian, German, French, Italian, Hindi, Bengali, Marathi, Punjab and Urdu. The Indio-European language family includes ten major branches. The English language comes under the Germanic Language branch. The two most spoken languages under the Germanic languages branch are English and German. It also includes a major

language which is the Dutch language. The Germanic branch consists of three branches which are the North, West and East Germanic language. The English language comes under the Anglo-Frisian branch under the West Germanic branch.

On the other hand, the Arabic Language descends from the Afro-Asiatic Languages family [26]. Figure 4-7 shows the origin of the Arabic Language. This family constitutes about 375 living languages spread over North Africa, the horn of Africa, Southwest of Asia, parts of Sahel and East Africa. The Afro-Asiatic family consists of five branches. It includes several famous ancient languages such as Ancient Egyptian, Biblical Hebrew and Akkadian. The Arabic language is the most spoken language in the Semitic group [25]. The Amharic language is the second most spoken language after Arabic in the Semitic family. Arabic is one of two branches of the Central Semitic language branch.

The difference in origin is reflected in vast differences between the two languages. Here we will ignore the differences to do with pronunciation and will consider only differences in grammar, which reflect differences in the text [22]. First of all, the Arabic language has 28 letters. Writing in Arabic is from right to left while in English it is vice versa. There are no capital letters in Arabic, but the form of the letter changes at the beginning, middle and end of a word. Punctuation is much more rigid in English than in Arabic. Arabic grammar is different from English grammar in some major aspects. Some of the main clear grammatical differences are as follows. Arabic nouns are either masculine or feminine, which affects the accompanying adjective. Arabic differentiates between male and female pronouns, verbs and words. Pronouns such as “they” and “you” have different forms for masculine and feminine, singular and plural. Tenses are different in the two languages. There is only a single present tense in Arabic, while English has simple and continuous forms. In Arabic there is no verb “to be” in the present tense. Arabic does not have phrasal verbs. There is no present perfect tense in the Arabic language, as in the English language.

All these differences affect the nature of the differences between Arabic text and English text. That is, the features which can be selected and used in the classification process to reveal the fingerprints of an author can be expected to be different in the two languages. For example, obviously, differences in the way two English authors use the different types of present tense cannot be useful features in Arabic because there is only one form of the present tense.

However, the so-called function words in English, although they cannot all be translated directly to Arabic to be used as a features, do often have counterparts in Arabic, which are words that play the same roles as the English function words. This is explored in the next chapter.

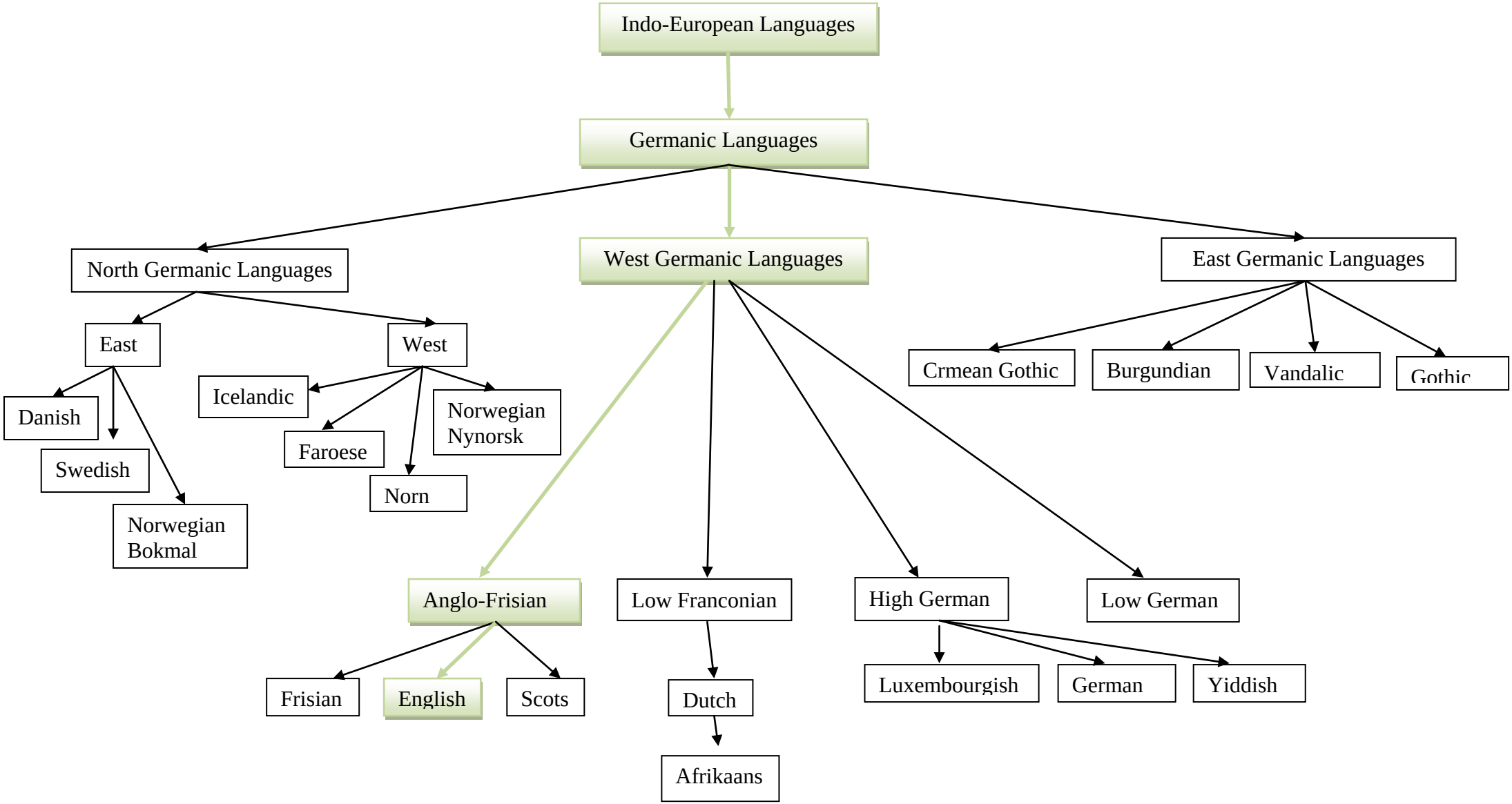


Fig.4-6. It shows the origin of the English Language.

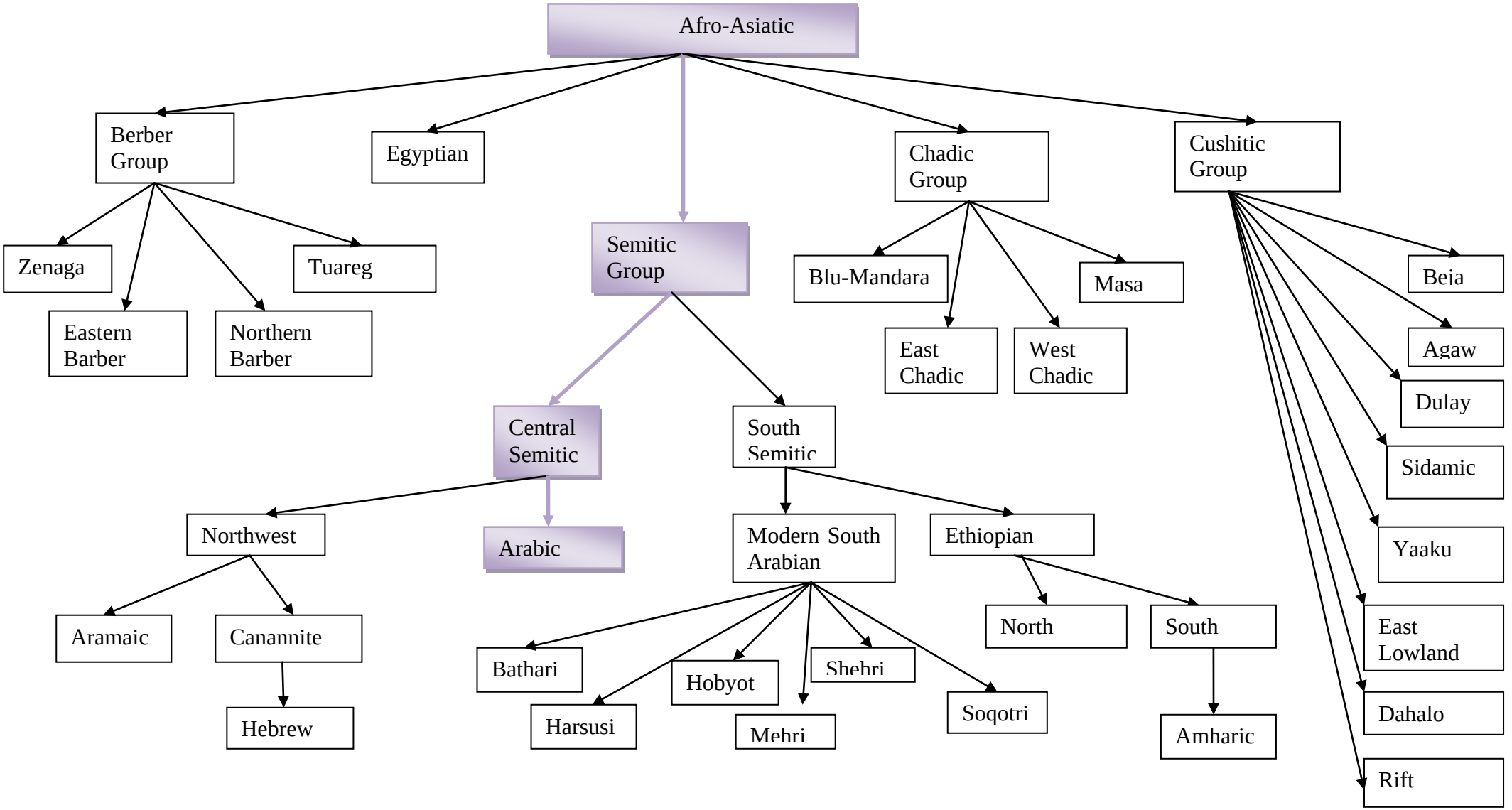


Fig.4-7 Shows the origin of the Arabic Language and how it descents from the Afro-Asiatic Family.

5

Arabic Function Words

Chapter overview

In this chapter we present investigations and experiments that lead to suggesting a set of Arabic function words, and then testing those function words in the context of authorship attribution. Section 5.1 is about finding and discovering the set of Arabic function words that seem usable as features for authorship attribution. It also introduces the steps towards settling on the best collection to be used in the experiments. Section 5.2 is about the Generalized Linear Model technique used as the classification Model in all the experiments. Section 5.3 explains our Evolutionary algorithm that searches for good subsets of features for classifying works to particular authors. Section 5.4 discusses several Experiments and shows their results. In Section 5.5 we discuss the factors that influenced the performance of classification, and we summarise in Section 5.6.

5.1 Arabic Function Words

Trying to find function words in the Arabic language was a challenge, after realising that the Arabic and English Languages are totally different. Having previously investigated in English Authorship Attribution using function words [27], we tried to find words that have a similar role to English function words but in the Arabic Language. Trying to find the 'same' words as in the English language was simply an unsuccessful idea. That is, the nature and the origin of the two languages are totally different, as explained in chapter 4. Even trying to translate the English function words into Arabic did not work, because the usage and role of these words are different from those in Arabic when translated. Moreover, more than one of the English function words when translated gave the same meaning in Arabic, and vice versa. Therefore, the criterion used in searching for Arabic function words was that they should resemble the English function words in terms of usage, not in terms of meaning. Thus, we sought words that are not nouns or verbs, and are important in linking words together and in helping to express the full meaning of the sentence. These words have grammatical roles and usages that can be expected to reflect different authors' styles, just as is the case for function words in English.

The number of words initially found of this type was a total of 106 words. These 106 words were a collection of prepositions, conjunctions, assertions, denials, interrogatives and pronouns. The pronouns are divided into demonstrative pronouns, relative pronouns, personal pronouns and possessive pronouns. In Arabic, the pronouns have many forms, the masculine and feminine for gender, singular, dual and plural for numbers, and 2nd and 3rd persons.

For example, consider Table 5.1, which shows a set of 65 Arabic function words chosen after some investigation that we will describe. The 1st person singular personal pronoun ("I", word 49) and plural form ("we", word 50) are as in English, while the 2nd person pronoun has a masculine form ("you", word 47) and a feminine form, but the latter is not in the table because it was properly eliminated from the list of 106 words. The masculine plural form ("you all", word 48) is retained. Also, the demonstrative pronoun ("this") has a singular masculine form in the table as word number 35, and a feminine singular form as word number 37, and also there is a dual masculine form and

a dual feminine form, both not in the table. However the masculine and feminine plural form (“they”) is the same for both genders, and included as word number 39.

Initial investigation with the full set of 106 potential Arabic function words was done by finding the frequencies of these words in the texts in our full dataset. Figure 5.1 summarises the results of that investigation, showing the average number of occurrences of each of the 106 words, averaged over the 14 books.

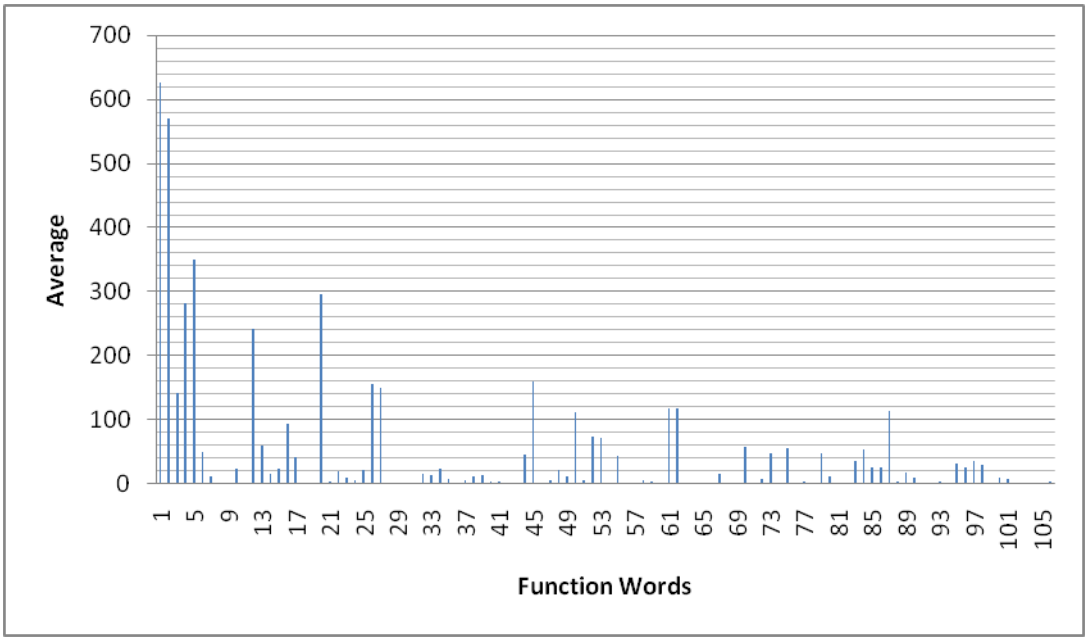


Figure-5-1 The average occurrences of the 106 potential Arabic function words in the full dataset. Some of the candidate function words did not appear in any of the texts such as words number 103,104 and 105.

Based on the findings of this initial investigation, 41 words could be eliminated. These were words that were scarce or absent from our dataset, reflecting their insignificant role and low usage, suggesting they are quite unlikely to be useful as features. These 41 words were eliminated with threshold of occurrences less than 90 words in all the texts or it didn’t occur in any of 6 texts or more. 12 out of the 41 words did not appear in any of the dataset texts. The frequencies of these words will be zeros in nearly all vectors so there is no point of adding them. There are two reasons that might explain the low usage of the 41eliminated words. The first reason involves diacritics. These are certain marks and signs added to a word, changing their form and case because of certain grammar rules, leading to change of the pronunciation of the words which leads to change in meaning. For example, the word “you” in the Arabic language could be masculine or feminine. The only difference between the two words is the diacritic. These diacritics could easily be omitted in some of the texts because readers could identify the intended

meaning of the words from context. The second reason is that all of the grammar rules of the Arabic Language come from the holy Qur'an, so usage of some of the function words are hard for writers to include or use in their text and books like in the holy Qur'an. This is because the holy Qur'an is words of god which is considered a miracle. One of the challenges of all times is that no one can produce or imitate even one verse of the Qur'an. Therefore, usage of some of the vocabulary and words are hard for writers to use in there text like God did in Qur'an.

The initial elimination of words left us with 65 words, each of which was used in every text. Table 5-1 presents those words, along with a translation into the English word that most closely reflects its usage.

1 في	2 من	3 عن	4 على	5 إلى
in	From	About	Over / on	To / towards
6 حتى	7 فلا	8 منذ	9 لا	10 ثم
until	Not	Since	No	Then
11 بل	12 لكن	13 أو	14 أم	15 أن
	But	Or	Or	That
16 كأن	17 إن	18 إذن	19 كي	20 لن
as if	The	Therefore	So that	Not
21 لم	22 ما	23 أي	24 ألا	25 أما
no	What	Any	Not	
26 ها	27 إذ	28 إذا	29 لو	30 لولا
		If	If	
31 هل	32 يا	33 نعم	34 بلا	35 هذا
Is?	Oh	Yes	Without	This
36 ذلك	37 هذه	38 تلك	39 هؤلاء	40 أولئك
that	This(feminine)	Such	They	Those
41 الذي	42 التي	43 الذين	44 هو	45 هم
whom	Which	Whose	He	Them
46 هي	47 أنت	48 أنتم	49 أنا	50 نحن
she	You	you all	I	We
51 الآن	52 بين	53 هنا	54 هناك	55 كان
now	Between	Here	There	Been
56 ليس	57 أصبح	58 ظل	59 ماذا	60 لماذا
not	Became	Keep	What	Why
61 كيف	62 كم	63 أين	64 متى	65 مهما
how	how many	Where	When	Whatever

Table 5-1 shows the 65 function words in Arabic and their translation in English. The highlighted ones are the words omitted in latter stages.

Following this, further analysis was done of the 65 words, in which I visualised the average frequencies of the function words across the different authors' texts to see if any significant variation or absence of variation was apparent. This led to the discovery that 11 of the words tended to have the same average frequencies across all the text, so they could be considered as contributing noise rather than contributing significant information that may help discriminate between different authors. These 11 words were

eliminated, and tests (as we shall see) generally found that performance of authorship attribution was improved. This group of 54 words is indicated in Table 5-1 by highlighting the 11 words that were eliminated due to low variance n their frequencies across the texts.

5.2 Linear Discriminant Analysis

The authorship attribution problem is basically a problem of classifying a text, where the classification comes from a small set of possible labels, usually two possibilities, where the label indicates the predicted author of the text. Given the relatively high dimensionality of data (the number of function words) compared to the number of data points available in each case, and also given the unbalanced class sizes we chose to use Linear Discriminant Analysis (LDA) for the classifier. LDA naturally and appropriately handles the problems of unequal class sizes. It works simply by finding a linear function of the data vectors that defines a separating hyper-plane which separates the data as well as possible, specifically aiming to minimise the ratio of within-class variance to between-class variance. The weights for the discriminating hyper-plane are learned by minimizing the cross-entropy error function. Meanwhile, to promote good generalization performance, we use leave-one-out cross-validation, and weight-decay regularization. Weight decay regularization attempts to keep low the absolute values of the weights in the discriminant function, which in turn tends to be associated with better generalization performance. But weight decay regularization involves a parameter which is difficult to choose correctly in advance. We thus do different experiments where we repeat the LDA training process several times for different values of this parameter.

Our LDA training implementation uses the Netlab library. In Netlab the model used for LDA is the Generalised Linear Model (GLM). The GLM calculates a linear combination of the input variables, in which the coefficients are the parameters of the model, and the result is followed by an activation function appropriate to the type of data being modelled [28].

The model works as single layer feed-forward network, and more detail is as follows:

The input values in our case are the vectors of function words frequencies. We assume there are d function words, so the vectors are d-dimensional. The input is

denoted x_i where i=1,...,d. The network forms c linear combinations of these inputs, where c is the number of outputs, to give a set of intermediate variables i,

$$a_j = \sum_{i=1}^{d} w_{ji}\, x_i + b_j \qquad \text{j} = 1, \ldots, \text{c},$$

with one variable a_j associated with each output unit. w_{ij} represents the elements of the weight matrix and b_i are the bias parameters.

The variables a_j are then transformed by activation functions of the output layer to give output value y_i. In our project we use the independent logistic sigmoid activation function applied to each of the outputs independently:

$$y_j = \frac{1}{1 + exp(-a_j)}$$

When using the logistic activation function for solving two-class classification problems (as in our case, where we are trying to classify a disputed text to one of two authors) there is the 'cross-entropy' error function which is

$$\text{E} = -\sum_{n}\sum_{k=1}^{c}\{t_k^n\, ln\, y_k^n \; + \; (1 - t_k^n) ln(1 - y_k^n)\}$$

To prevent any weights becoming too large, a weight decay penalty term is added to the error function, which becomes

$$\text{E}_r = \text{E} \; + \; \alpha \sum w_i^2$$

Where α is non-negative regularisation parameter. The optimal value of α is usually found by cross validation. Therefore, Leave-one-out cross-validation is performed to find the best regularization parameter α by evaluating the error E through training using a range of different α values from $1.0e^{-10}$ to $1.0e^{2}$. The best α value is the value which gives the smallest E.

Training is done via the Iterated Re-weighted Least Squares algorithm, using default parameters in the Netlab implementation. This training process is repeated for a range of different values of α, as mentioned earlier.

Then an unseen set of test data is used to estimate the performance of the trained classifier, using the value of α that was best during training.

Further detail on finding the best value for the regularization parameter α is as follows. Testing a small number of values (e.g. 10) means that the training process is

much faster, however testing more values (e.g. 20) means there is a better chance of finding a good choice of the parameter. I tested 10 and 20 logarithmic values for α from $1.0e^{-10}$ to $1.0e^{2}$. I used the matlab function called logspace to produce logarithmic spaced points from $1.0e^{-10}$ to $1.0e^{2}$. The 10 values used for the regularization parameter α are in table 5-2 and the 20 values used are in table 5-3.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1.00e-10	2.15e-09	4.64e-08	1.00e-06	2.15e-05	0.00046	0.01	0.215	4.642	100

Table 5-2. This table has the 10 values of regularization parameter α

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1.00e-10	4.28e-10	1.83e-09	7.85e-09	3.36e-08	1.44e-07	6.16e-07	2.64e-06	1.13e-05	4.83e-05
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
0.0002069	0.00089	0.0038	0.0162	0.0695	0.2976	1.2743	5.456	23.357	100

Table 5-3. this table has the 20 values of the regularization parameter α

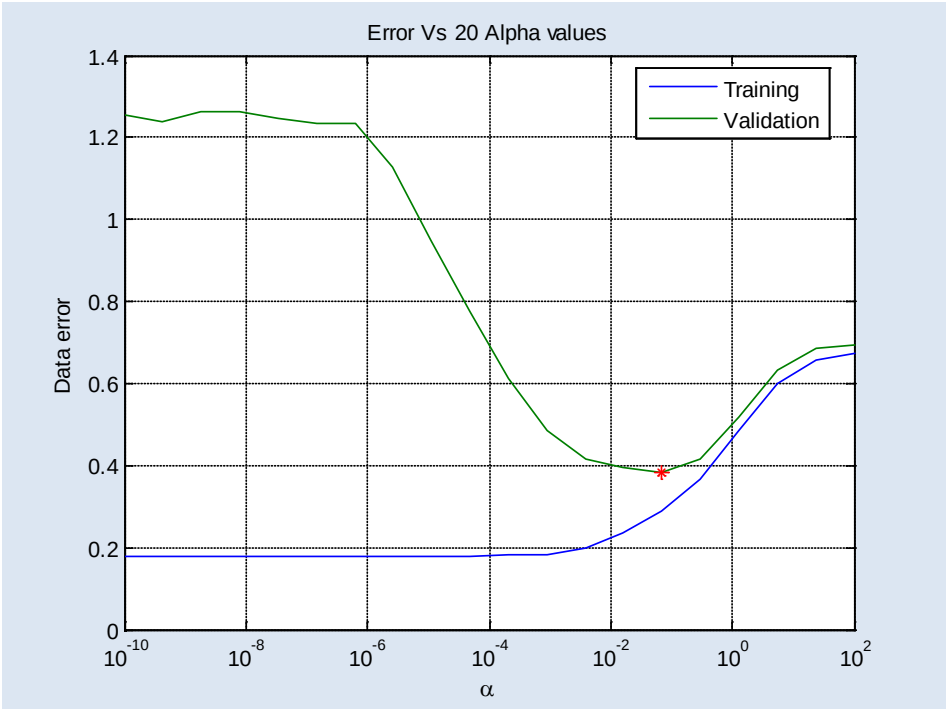


Fig. 5-2 This graph shows error values versus 20 regularisation parameter values. The red asterisk or star shows the best α value reached in the validation dataset. This α value when used gave the least error value which is 0.384.

To illustrate the difference between using 10 and 20 α values we plotted two graphs to show how different values of α could lead to better validation data error values. In fig.5-2 we plot error values versus 20 α values and in fig.5-3 only 10 regularization parameter values are used. It can be inferred from the two graphs that 20

α values led to the best error value, which is 0.384 at α value 0.0695 which is value number 15 in table5-3. In the tests with 10 values the best was 0.01 with error value 0.0399. In this particular example, the 20 values test performed better, but this is not the case on all other tests.

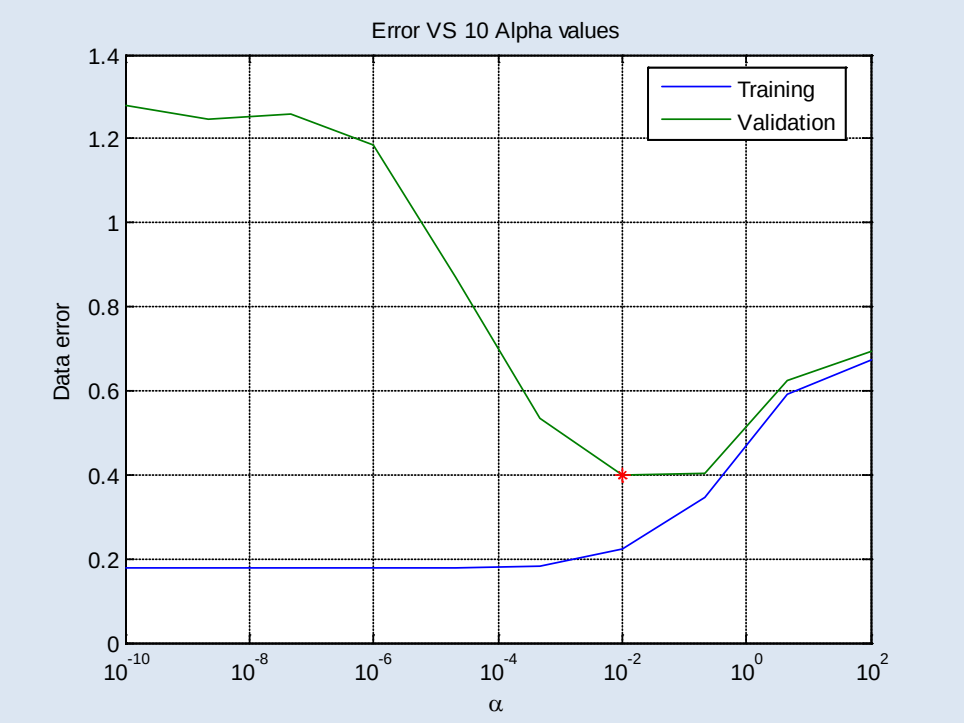


Fig. 5-3 This graph shows error values versus 10 regularisation parameter values. The red asterisk or star shows the best α value reached in the validation dataset. This α value when used gave the least error value which is 0.4.

5.3 Evolutionary Algorithm for Feature Subset Selection

In our approach, the Evolutionary Algorithm (EA) plays an important role in selecting subsets of function words to be trained by the GLM. It starts with a small population of five unique candidates; each candidate is a singleton chromosome which consists of one function word, chosen randomly from the 65 function words or the 54 function words, depending on the experiment. Every one of the candidates in the population is trained by the GLM and the output of the GLM is evaluated using the Area Under Curve (AUC), where the curve in question is the Receiver Operating Characteristic (ROC) curve. The AUC is the fitness value used to sort the population from best to worst. Binary tournament selection is used to choose a single parent, and the chosen parent is mutated by randomly choosing either of: deleting, adding or replacing one of the features in the chromosome. But there are some restrictions in doing the mutation. First,

a singleton chromosome can't be deleted; second, adding a new feature should not duplicate one of the existing features in the chromosome; third, in producing new offspring by deleting, adding or replacing one feature, the new offspring should be unique in the population. Then, the mutant is evaluated and enters the current population if it is better than the worst candidate. The population is then sorted according to the fitness value from best to worst. In some cases while comparing between the new offspring and one of the parents, the fitness values in both are equal; in this case we choose the candidate with the smallest number of features.

5.3.1 The Algorithm

The whole algorithm works as shown in flowchart in fig 5-5 as follows:

The first step is the creation of the population as in fig 5-6 from line number 2 to line number 10. The population consists of a fixed number of candidates. Each candidate of the population contains only one function word. The function word is picked randomly from the pool of the 65 or the 54 function words. We ensure that every candidate is unique in the initial population, and we continue to promote uniqueness by comparing every new offspring candidate with the rest of the current population, regenerating a new offspring candidate if it is not unique.

In the next step as in flowchart 5-5, we evaluate all the candidates in the population and sort them according to the fitness function from best to worst fig 5-4 shows a graphical illustration on a ROC curve how the chromosomes are evaluated. This is done by training each candidate with the LDA. Then, the candidate is tested on unseen test data. Next, the candidate fitness is evaluated by finding the area under the ROC curve for the GLM classifier built for that candidate as in fig 5-6 from line 11 to line 14.

Next, the algorithm enters a loop of fixed iteration number of either 100 or 300 iterations as in fig 5-6 line 15. Then binary tournament selection is used to choose one of the candidates to be mutated as in fig 5-6 from line 16 to line 24. The candidate is mutated by choosing at random one of the eligible mutation methods. If the addition method is chosen, just one more function word is added to the current chromosome. If replacement is chosen, one of the genes, which is a function word, of the chosen chromosome is selected randomly to be changed. But if deletion is chosen, then as shown in flowchart 5-5, the algorithm makes sure that the candidate is not a singleton. This is required to prevent deletion of the only single gene in the chromosome and

creates empty chromosome which is not valid. Next, the new offspring is tested for its uniqueness as in fig 5-6 from line 25 to 31. That is, the algorithm makes sure the new offspring is not duplicated or not identical to an existing candidate in the population

Finally, the offspring is evaluated and compared to the worst candidate in the population as in fig 5-6 in line 32 and 33. If it is better than the worst it replaces it. Then as shown in the flowchart 5-5 the population is sorted again and repeats the whole process till we reach the required number of runs.

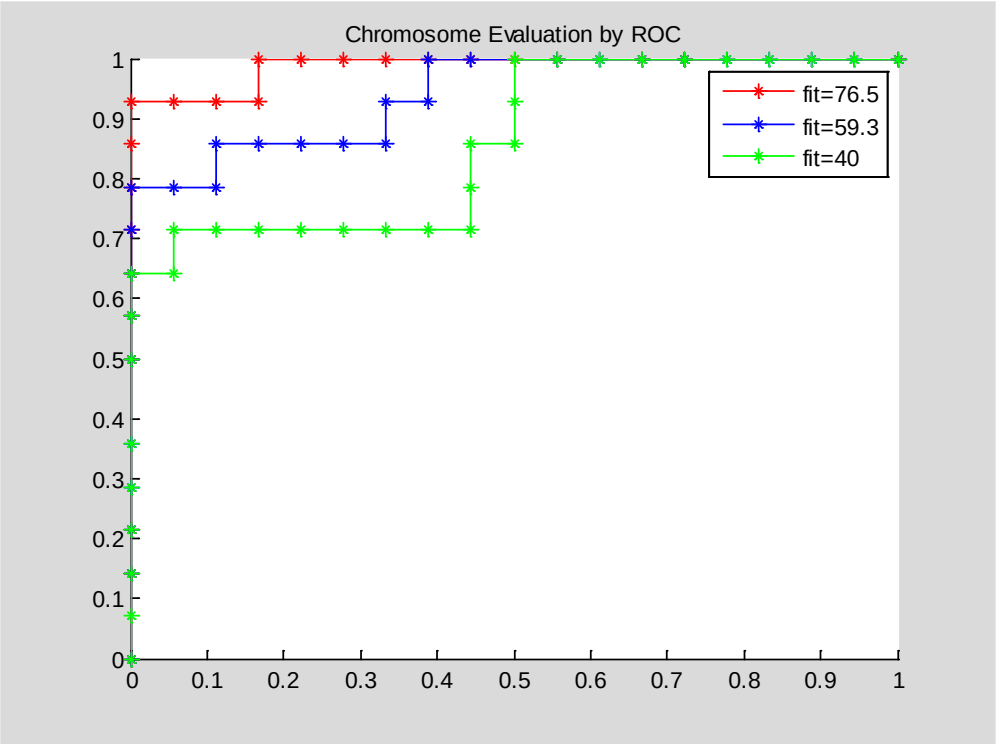
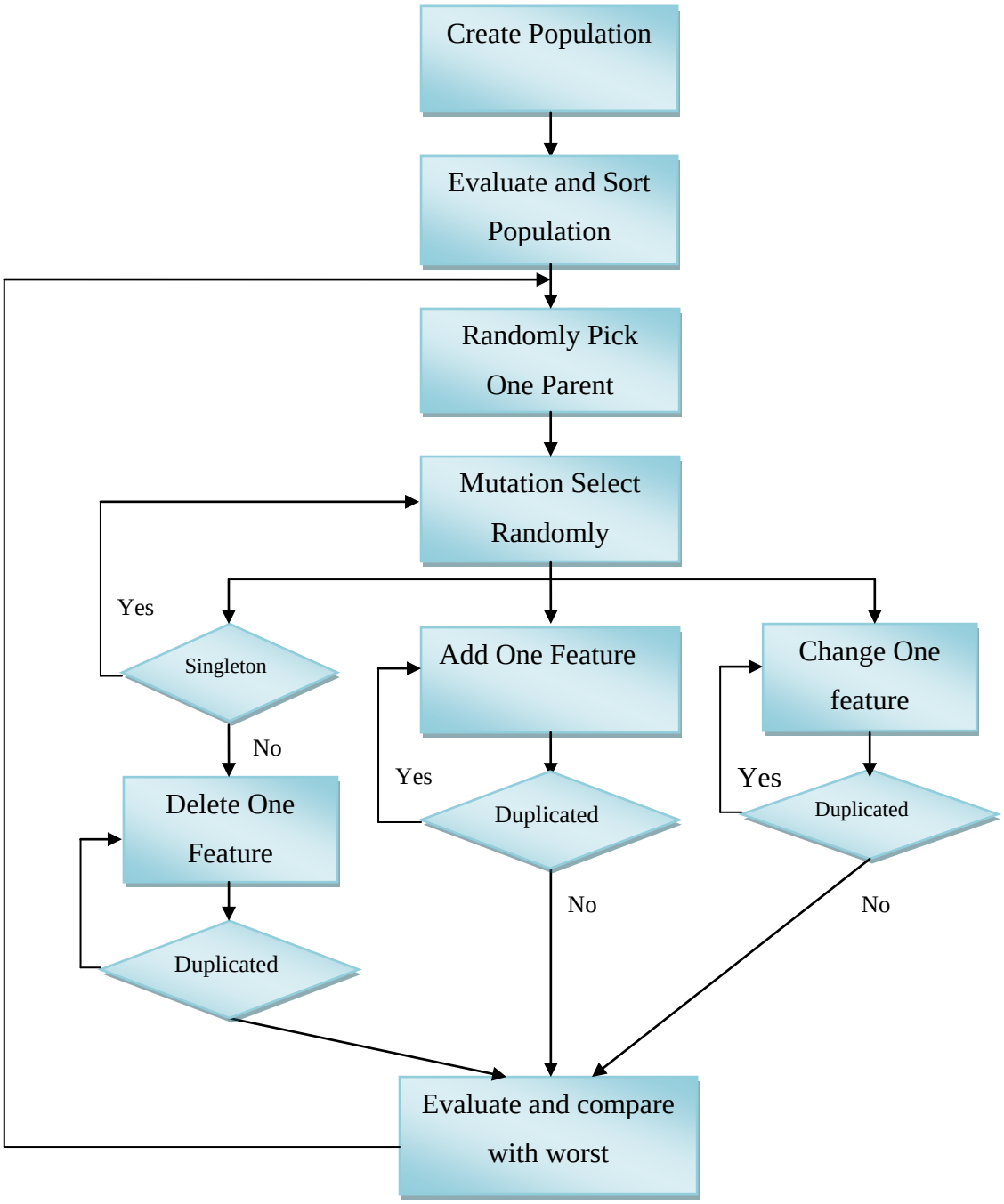


Fig.5-4.This is an illustration of how a chromosome is evaluated. There are three different ROC curves in this figure. Every line represents a chromosome evaluated according to performance in training. The worst chromosome is the green line with fitness 40. The best chromosome is the red one which scored in the training a fitness of 76.5.



Flowchart 5-5 shows the flowchart of the evolutionary algorithm.

```
Evolutionary Algorithm
1: start
2:        for pop=1 to populationSize
4:           F = Select_Feature ( 65 or 54 features);
5:           Boolean= Unique(F)
6:           if (1)
7:              population(pop)=F
8:           else
9:              F=Select_Feature
10:          end
11:          Alpha = Train_LDA( F );
11:          Classify_LDA ( F , Alpha);
12:          Evaluate(F)
13:       end
14:       Sort(population)
15:       for i = 1 to N
16:          F= Select Randomly Features of population
17:          R = Select_Randomly;
18:          if ( R = =1 )
19:             F = Add_Feature;
20:          else if ( R = = 2)
21:             F = Remove_Feature;
22:          else
23:             F = Replace_Feature;
24:          end
25:          Boolean=Unique(F)
26:        if (0)
27:          goto 18
28:        else
29:            Alpha = Train_LDA( F );
30:            Classify_LDA ( F , Alpha);
31:        end
32:        Evaluate(F)
33:         Sort(population)
34:       end
35: end
```

Fig 5-6 the Evolutionary algorithm

5.4 Authorship Attributions Experiments Using 65 and 54 Function Words

The dataset consists of 14 books as we described before in chapter 4. Six of the books were divided into two parts because they were very big. The largest book has 59,116 words and the average number of words in these 6 books is 49,617. The reason behind dividing these books was to ease the processing task of eliminating unwanted characters and blank spaces. Another reason is to avoid or minimize the problem of imbalanced class sizes. That is, using these books without dividing them, leads to the numbers of data vectors for the large books in some cases being double or triple the numbers of data vectors of other books, which in turn leads to unbalanced class sizes.

Nevertheless, one of the advantages of using the linear discriminant analysis classifier is that it could help to overcome this problem.

In table 5-4 the books and authors are given IDs to help in the presentation of results. The table also shows the number of words of each book. The books which were divided are shown with the two parts and their numbers of words.

Authors	Book details and Ids
Ibrahim Khalil	Haris Al Maiz (**HAM**: 14,679 words)
	Sodom Sebake Al Awez (**SSAA**: 2 parts, 28156 words and 29518 words)
Basem Ibrahim Abdo	Gesr Al Mawt (**GAM**, 37567 words)
	Zahra fi Al Remal (**ZFAR**: 2 parts, 23241 words and 26061 words)
Taleb Omran	Ahzan Al Sinbad (**AAS**, 20389 words)
	AlBood Al Khamis (**AAK**, 13987 words)
	Madina Kharig Al Zaman (**MKAZ**, 14513 words)
	Al Fetiah Al Aghrar we Asfar al Kashf (**AFAA**, 19063 words)
Mary Show	Defly (**DE**, 24062 words)
	Awel Hob and Akheir Hob (**AHAH**: 2 parts, 18848 words and 19807 words)
Mohamed Youssef Salibi	Al Taih (**AT**, 36892 words)
	Sebahaa fi Al Wahl (**SFAW**: 2 parts, 25647 words and 28274 words)
Hessen Abd Al Kareem	Al Nabaa (**AN**: 2 Parts 29644 and 29472 words)
	Shagaret al Toot (**SAT**: 2 parts, 19024 words and 19998 words)

Table5-4. This Table shows the Authors and their books which are used in the experiments.

First of all, after preparing and processing the entire texts to be ready for experiments as explained in chapter 4, I created five test cases. Each case contains four books belongs to two different authors (each author has two books). One book each is for the training and also one book each is for testing. When books are divided, the different parts do not play a part in the same test case. Also, books that are both used in one test case are not both used in any other test case.

In table 5-5, there are details of the five test cases named A, B, C, D and E. The books of each test case are stated with their IDs as explained in the table 5-4.

Test case	Train and test set details
A	Training set: Books **AAK** and **HAM** Test set: Books ***MKAZ*** and ***SSAAp2***
B	Training set: Books **AHAHp1** and **SATp1** Test set: Books ***DE*** and ***ANp1***
C	Training set: Books **GAM** and **SFAWp2** Test set: Books ***ZFARp1*** and ***AT***
D	Training set: Books **ANp1** and **AAS** Test set: Books ***SATp2*** and ***AFAA***
E	Training set: Books **AFAA** and **ZFARp2** Test set: Books ***MKAZ*** and ***GAM***

Table 5-5. shows the test cases names and the books used in each case.

A reminder of the basic setup for each experiment is as follows. If there are two books in the test set, each book is divided into contiguous pieces of text of *N* words each, where *N* is the chunk size used for the experiment. So, if a book has 20,410 words, and we are using chunk sizes of 1000 words, then there will be 21 chunks. Each chunk is transformed into a data vector, which contains the frequencies of the 65 or 54 function words in that chunk, and where the class value is the author of the book involved.

To obtain reliable results I did more than 17 experiments. Every experiment uses the five different cases as mentioned before named A, B, C, D and E. We did 10 different runs of the EA/GLM approach in every case, which means 50 runs in one experiment. Every single run was done twice, once for 100 iterations of the EA/GLM algorithm, and once for 300 iterations of the EA/GLM algorithm. Shorter runs would be expected to avoid overfitting, however it can also be said that the use of the GLM approach and regularization was appropriate to avoid overfitting. Nevertheless it was worth investigating whether the number of iterations used to develop the feature subsets had an important effect. Overall, the processing requirements in every experiment were extensive, and the average run time for one run out of 50 runs required for every case took about more than one hour to complete as shown in table 5-6. The overall processing time of the 12 experiments as shown in table 5-6 took more than 722 hours to complete. There were two computer used in carrying out the experiments. One was of dual core processor T7200 2.0GHz and 2 GB DDR2 RAM and the other one was dual core processor T9900 3.06GHz and 8GB.

The experiments were organized into three main groups, which relate to the chunk sizes used: 1000, 2000 and 3000 words chunk sizes respectively. For each chunk size, experiments were done on each of the five test cases, for each of the two function words sets (54 words and 65 words), and for either 300 iterations, or for both 100 and 300 iterations. E.g. in the case of the chunk-size 3,000 words and 100 iterations, it was observed during the 300 iterations runs that it would not be worth examining the 100-iterations results in this case.

Experiment Name	Total run time (10 runs)	Average run time
Combined 54 FW of 1000 and 2000	122:47:22	02:27:21
54 FW of 2000 words 300 iterations	38:33:09	00:46:16
54 FW of 2000 words 100 iterations	29:03:27	00:34:52
54 FW of 3000 words 300 iterations	15:58:09	00:19:24
65 FW of 2000 words 300 iterations	57:15:38	01:10:07
Combined 54 FW and Letters 300 iterations	89:05:36	01:46:55
65 FW of 1000 words 300 iterations	97:05:23	01:56:30
65 FW of 1000 words 100 iterations	30:31:58	00:36:38
54 FW of 1000 words 300 iterations	64:37:53	01:17:33
65 FW of 3000 words 300 iterations	33:09:59	00:39:48
5000 characters 300 iterations	88:39:24	01:46:23
10000 characters 300 iterations	56:01:01	01:09:20
Total	**722:48:59**	**01:12:36**

Table 5-6. this table shows the processing time of the experiments and the average processing time of every run.

5.4.1 Result Calculation Methodology

First of all I would like to show how we record the results, to avoid confusion in later sections. Every experiment is carried out on the five cases. Every case is re-run 10 times independently. In every run, 100 or 300 iterations are done to find the best features that can classify the test books. The result of every run is a number of best-performing chromosomes. Then, a test is carried out on these best chromosomes using the test books in every case. Test book1 always refers to the book of the first writer and test book2 refers to the second writer. Finally, the average performance of the ten tests is recorded for every case. For example, table 5-7 shows the results of the ten tests of case C in one of the experiments.

Run	Test Book1 Performance %	Test Book2 Performance%
Run1	86.9565	70.9259
Run2	79.5396	53.8462
Run3	79.4050	93.8131
Run4	88.1007	78.9474
Run5	95.3416	42.8571
Run6	81.4010	90.6863
Run7	75.6522	59.3434
Run8	78.5326	78.2118
Run9	98.0435	75.6173
Run10	82.4534	90.1786
Mean	**84.5426**	**73.4427**

Table 5-7 this table shows the average performance in every Run and the mean value of all 10 runs.

For simplicity I will only present the mean of these 10 results in the tables of every case. Then the final output performance of case C will be like this:

Case C	78.22

The performance is calculated by the percentage of how many chunks of the test books is identified correctly by the best chromosomes produced out of training. That is, for example if the test book is divided into 10 chunks and the best chromosomes (of one run) classified correctly 8 out of the 10 chunks then the performance percentage is 80.00.Then calculate the performance percentage for the ten runs of every case to produce the overall performance percentage as shown in table 5-7.

Finally, in selected cases when we analyse the results we perform a T-test. This is always a 1-tailed T-test assuming unequal variance, testing whether a finding (e.g. that 54 function words is better than 65 in a particular context) is statistically significant. In these cases we usually set our threshold for significance at 0.9 or 90%.

5.4.2 Experiments with 1000 words chunk size

For this chunk size I did tests with the 54 function words and the 65 function words sets. In the 65 function words set I did two tests, one with 100 iterations and the other with 300 iterations. Based on the results of the 300 iterations only was used with the 54 function words group.

Table 5-8 shows the result of the five test cases performance in classifying the test books using 65 function words and 100 iterations. First, all the values in the table are mean values of the ten runs. As explained in result calculation methodology, each value in the performance column of Table 5-8 is the mean of 20 numbers – the 10

values for performance on test book 1and the 10 values for performance on test book 2. As we can see, the overall mean is 84.26%.

65 words and 100 iterations	
Case Name	Performance
A	77.558195
B	89.796715
C	78.99266
D	85.781865
E	89.16863
Total Mean	**84.259613**

Table5-8 shows performance of 65 function words using 100 iterations in 1000 words chunks

Table5-9 shows the results of the five test cases using the 65 words with 300 iterations. There is no significant improvement in the total performance mean between 100 and 300 iterations in the 1000 chunk size.

65 words and 300 Iterations	
Case Name	Performance
A	76.104535
B	90.116225
C	80.187335
D	86.43699
E	89.82798
Total Mean	**84.534613**

Table5-9 shows performance of the 65 words using 300 iterations in 1000 words chunks

Table 5-10 suggests that using the 54 function words set is better than using the 65 function words set, at least when chunk sizes of 1000 are involved. For each of the five test cases, a one-tailed paired T-test comparing the 54-word results with the 65-word results in the 1000-word chunk / 300 iterations context. In two of those five cases, 54-words is found better with statistical significant (99.3% and 93%); in another two cases, 54-words has a better mean, but the result is not significant, and in the final case the 65-words has a better mean but the result is not significant.

54 words and 300 iterations	
Case Name	Performance
A	87.60815
B	90.4939
C	78.83095
D	86.9797
E	84.6044
Total Mean	85.70342

Table5-10 shows the performance of 54 words using 300 iterations in the 1000 words chunks

5.4.3 Experiments with 2000 words chunk size

In the 2000-chunk-size experiments we also investigated the difference between 10 and 20 trials for finding a good regularization parameter α. These tests were done on the 54-function words tests. Table 5-11 shows that the classification performance improved when using 20 α values, in both the 100 and 300 iterations runs.

54 words and 100 iterations			54 words and 300 iterations	
Case Name	**10 Alpha**	**20 Alpha**	**10 Alpha**	**20 Alpha**
A	81.50295	88.21895	86.9065	93.80055
B	89.2221	87.9592	89.46525	86.2738
C	81.96735	82.6434	82.8371	82.85175
D	86.6019	88.08315	84.67765	85.2139
E	89.4849	89.3882	88.481	90.20645
Mean	85.75584	87.25858	86.4735	87.66929

Table5-11 shows the performance of 54 words using the 100 and 300 iterations. It also shows the difference between using 10 and 20 values of α. In this case the 20 values performed better.

Figure 5-7 summarises this graphically, showing the difference in performance due to increasing the number of α values, which increases the chances of getting better error values when training with the GLM as discussed before.

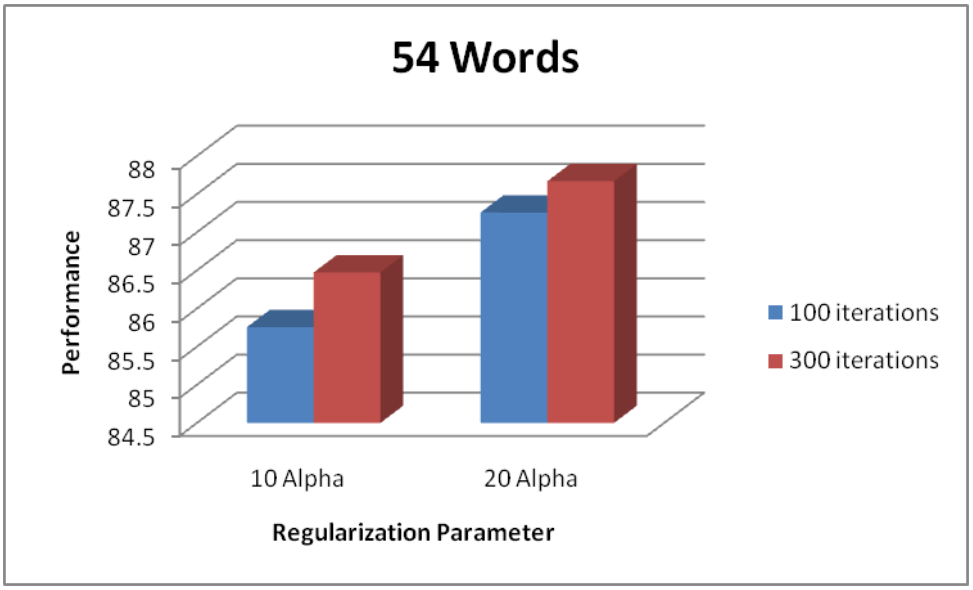


Fig 5-7 this graph shows performance of the different values of α using 54 words. The 20 values search performed better than the 10 values search in both 100 and 300 iterations.

The performance of the 65 function words group, summarised in Table 5-12 and Fig 5-8, also show that 300 iterations led to better results than 100 iterations. The difference is not big but it shows that the more the number of iterations is the bigger the chance of finding better features.

For each of the five test cases, a one-tailed paired T-test comparing the 54-word results with the 65-word results in the 2000-word chunk / 300 iterations context. In one of those five cases, 54-words is found better with statistical significant (93%); in another three cases, 54-words has a better mean, but the result is not significant, and in the final case the 65-words has a better mean but the result is not significant.

65 words and 10 Alpha		
Case Name	**100 iterations**	**300 Iterations**
A	87.48795	87.736655
B	86.544325	89.490635
C	78.893675	82.12723
D	84.928545	84.56763
E	87.36235	88.38084
Mean	85.043369	86.460598

Table5-12 It shows the 300 iterations is better than 100 iteration in 65 words and using 10 α values in the 2000 words chunks.

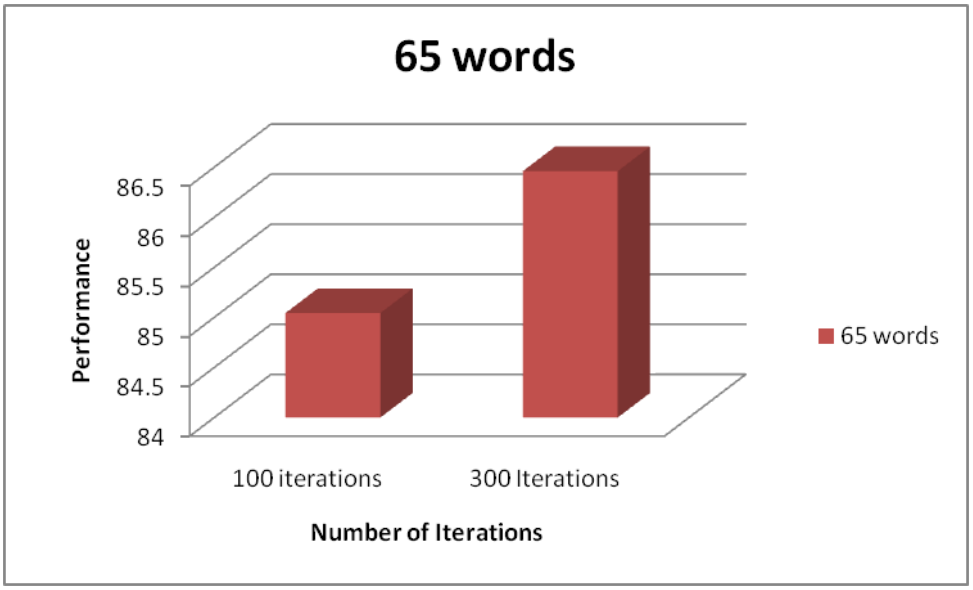


Fig5-8 it shows that the performance of 300 iterations is better than 100 iterations using 65 function words in the 2000 words chunks.

5.4.4 3000 words chunk

In this case only 300-iterations results are shown in table 5-13. As in the experiments with other chunk sizes, the 54 words group performed better than the 65 words group, but in this particular chunk size the difference in the performance was greater than any one of the other experiments as in figure 5-9.

300 Iterations		
Case Name	**54 Words**	**65 words**
A	88.5822	73.7333
B	88.4556	88.7663
C	89.0204	87.1512
D	79.86025	82.46405
E	86.8617	87.8107
Mean	86.55603	83.98511

Table5-13 Shows the results of the experiments of the 54 and 65 function word in the 3000 words chunks. The 54 function words performed better than the 65 function words.

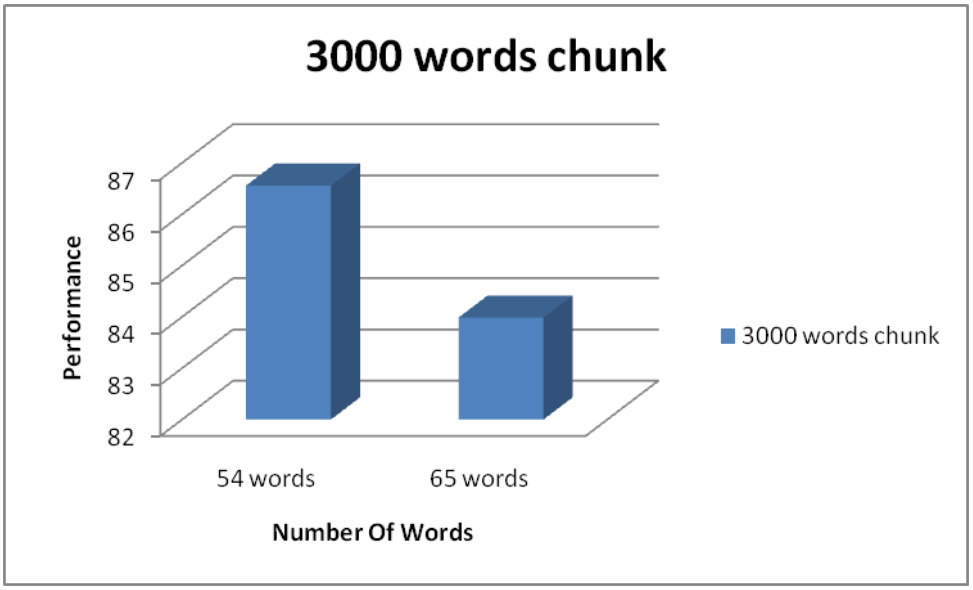


Fig5-9 shows the difference in performance between the 54 function words and the 65 function words in the 3000 words chunks.

For each of the five test cases, a one-tailed paired T-test comparing the 54-word results with the 65-word results in the 3000-word chunk / 300 iterations context. In one of those five cases, 54-words is found better with statistical significant (99%); in another three cases, 65-words has a better mean, but the result is not significant, and in the final case the 54-words has a better mean but the result is not significant.

5.5 Discussion Of Results

The first and most interesting factor that affects the results of authorship attribution studies in these experiments is the choice of function words set (54 and 65) – this is also basically the choice of features to use in order to represent an author's style. One question that we ask, and test with the experiments in this chapter, is which of the two function words sets are better for reflecting an author's distinct writing style and extracting his or her 'fingerprint'.

The second factor of interest is the chunk size. Choosing the proper chunk size is one of the important factors because it affects the amount of knowledge or data related to an author's fingerprint that can be included in one vector. Too small a chunk size seems like a bad idea, because there are not enough words to capture the authors' style, and we can expect the results to be noisy. Too large a chunk size means that we would have a reduced set of data vectors, and so any machine learning method would find it hard to grasp reliable patterns.

A third factor is the number of iterations of the EA/GLM program. We generally found that 300 iterations outperforms 100 iterations, and our experiments were limited in exploring this aspect. But what is of interest here is finding that a relatively small number of iterations within the algorithm can give acceptable results, because this means that authorship attribution using this method is not too onerous.

A fourth factor, which we also look at briefly, is the class size, which is the number of vectors of each class in the training data every class. Although this is to some extent tied together with chunk size, the question here is whether short books can be useful in training data for authorship attribution.

5.5.1 Function words sets

I used two groups of function words in the research to find the best one to classify with. One group contains 65 function words, and the other group contains 54 function words. The 65 function words group is the result of eliminating 41 words of rare or scarce presence in all texts from the original 106 words group. One clear finding is that the use of an appropriate function words set is suitable for authorship attribution in Arabic. We can conclude this because the results we have seen so far are generally competitive with results of authorship attribution studies in other languages, reported in the literature review chapter. Also, this represents the first time that function word sets have been proposed for Arabic. When comparing the 65-word and 54-word sets, we generally found that the 54 words set gave better performance. This implies that, the 54 words include many that different authors use in individual ways, reflecting their styles. This is also true of the 65-word set, which includes all of the 54-words, however this larger set seems to add words that only contribute noise and confusion to the classification, probably because different authors do not use them in different ways, or perhaps use them inconsistently.

We can see the difference in performance between 54 and 65 words illustrated in Figure 5-10, which shows the results for each chunk size. It shows that the eliminated 11 words from the 65 group improved the performance in all cases of chunk size. The statistical analysis of the results tended to support the conclusion that the 54-word set was more applicable.

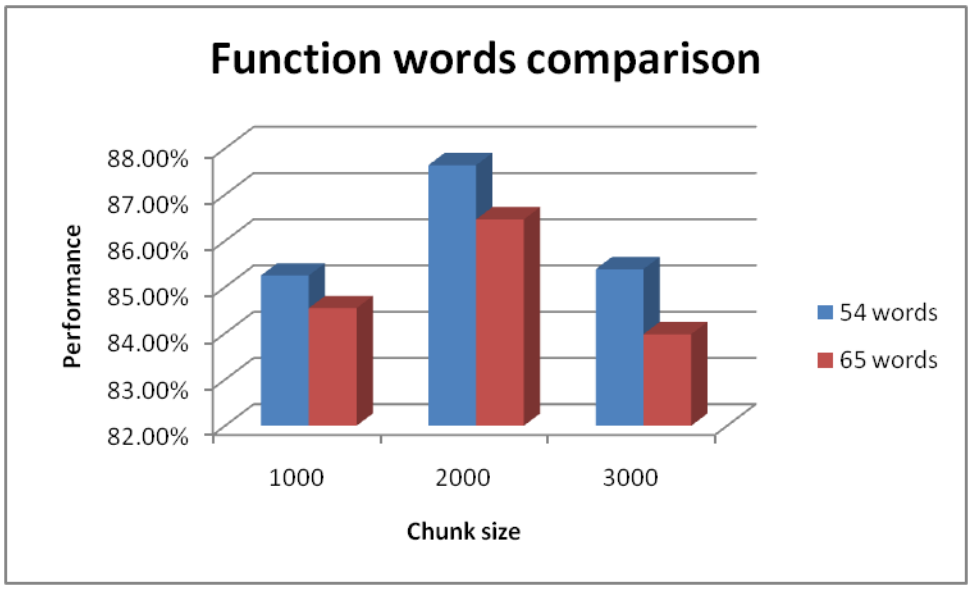


Fig 5-10 this graph compares among the performance of 54 function words and 65 function words in all words chunk sizes. It shows that 54 function words outperformed in all chunk sizes.

5.5.2 Chunk size

The chunk size is the number of words from the book for which each data vector is calculated, where a data vector contains the frequencies of function words in a single chunk. So, the text is divided into a number of chunks and the frequency of each function word is calculated in each chunk. For example, in a case which a text size is 13,221 words and the chunk size used is 1000 words, the text file will be divided into 13 chunks of 1000 words each and last chunk with only 221 words remainder is deleted. Then the average frequencies of the function words are calculated in each chunk. The output is a number of vectors, each vector represents one chunk. In this example the output is 13 vectors represent the number of chunks and each vector contains 65 values of average frequencies which represents the 65 function words.

The main chunks sizes used are 1000 words, 2000 words and 3000 words. One factor of importance is the knowledge contained in each vector. When chunks are 1000 words long, there is a large number of vectors, but each vector has less data or knowledge which may not reflect enough information about the author's style. On the other hand, larger chunks contain quite reasonable information about the text but the number of vectors becomes less which affects the training process of the classification tool.

Another factor in these results could be the data discarded due to the chunk size. The total number of words in the texts is 478,842 words. The discarded data are the

remainder that found in the last chunk which is smaller than the chunk size in use. The discarded data in the 1000 words chunk size experiments amounts to 9,842 words which is 2.1% of the text. In the 2000 words case the discarded data is 20,842 words, which is 4% of the data, and in the 3000 words chunk size cases the discarded data is 34,842 words which is 7% of the text. As we have already seen in Figure 5-10, using a chunk size of 2,000 words slightly improved the performance of classification but it is not significantly better than the other chunk sizes. It might be argued that 3,000 words chunk sizes may have suffered from the 7% loss of data, however we saw no evidence for such an effect as we moved from 1000 words to 2000 words.

5.5.3 Number of Iterations

The Algorithm starts with the creation of the initial population, and then runs for a certain number of iterations. The more the number of iterations, is the bigger the chance to find a good subset of features to classify with. First, I did runs of 100 iterations and compared them to 300 iterations. 300 iterations gave and found better feature subsets.

Table 5-14 is an example that shows the difference among the candidates in the population from the beginning of a run, after 100 iterations and again after 300 iterations. This is an example of a population of 5 candidates and it shows how poor were the fitness values in the starting population, and how much it developed to reach the best fitness, which is the value of 98. But in the two examples it can be seen that the number of the function words used in each candidate in the 300 iterations final population is less than the number that generally appear at 100 iterations.

In earlier work, not reported in this thesis, a similar EA/GLM approach was developed and used for authorship attribution in English, and the emphasis was on finding small sets of English function words that (in conjunction with the LDA process) were sufficient for discriminating in some famous dispute cases. Although in the current research I do not focus on this aspect, it is true to say that smaller subsets are probably better to avoid overfitting. So, even though 100-iteraton tests can find high performing chromosomes in the experiments here, as we see below, the 300-iteration tests give better performance probably because the number of function words in the chromosome tends to be smaller.

100 iterations	300 iterations
------main population-----	**-----main population-----**
Feature: (42)	Feature: (4)
Fitness: 1.190476e-001	Fitness: 0
----------	----------
Feature: (1)	Feature: (64)
Fitness: 1.134921e+001	Fitness: 6.349206e-002
----------	----------
Feature: (25)	Feature: (6)
Fitness: 3.350794e+001	Fitness: 1.224603e+001
----------	----------
Feature: (49)	Feature: (32)
Fitness: 5.300000e+001	Fitness: 5.701587e+001
----------	----------
Feature: (47)	Feature: (58)
Fitness: 7.730952e+001	Fitness: 7.250794e+001
------Final Population----	**-----Final Population----**
Feature: (18 21 37 47)	Feature: (13 32)
Fitness: 98	Fitness: 98
----------	----------
Feature: (28 32 42 63)	Feature: (13 27)
Fitness: 98	Fitness: 98
----------	----------
Feature: (14 28 32 42)	Feature: (13 32)
Fitness: 98	Fitness: 98
----------	----------
Feature: (21 28 32)	Feature: (7 13)
Fitness: 98	Fitness: 98
----------	----------
Feature: (28 32 42)	Feature: (28 58)
Fitness: 98	Fitness: 98

Table5-14 this table shows the difference in the final population of 100 iterations and 300 iterations. The more the number of iteration the better the final population is. The 300 iterations find less number of function words that reached the best fitness which are two word while the 100 iterations find 5 and three function words.

5.5.4 Class sizes

When constructing the five test cases, consideration was given to experimenting with imbalances in the class sizes in some of the training data by being careful about which books to pair together. I picked two cases in which the training books are of nearly the same number of words; these are test cases A and B. In test case A, book AAK is 13987 words and book HAM is 14,679 words. In test case B, book AHAHp1 is 18,848 words and book SATp1 is 19024 words. The number of vectors in the training data for these two cases will be nearly the same in each of the chunk size experiments. In test case E, there is a modest but not great imbalance, with book AFAA at 19,063 words and book ZFARp2 at 26,061 words. Meanwhile in test cases C and D, the difference between the books are large. In test case C, book GAM which is 37,567 words, is larger than book SFAWp2, which is 28,274 words. In test case D, book ANp1 is larger than book AAS by 9255 words. These two test cases will have ~9 vectors different between the two classes in the 1000 words chunks case and about 4 vectors difference in the 2000 words chunks case.

Of course, this imbalance in class size is not a decisive or critical issue in classification accuracy. There are no huge (e.g. orders of magnitude) differences, and also it is not the only factor that affects the classification process, but it was quite interesting to observe if this led to any effects. Also, it needs more experiments and further investigation to judge accurately. The results reflect that there is better classification in the test cases A, B and E, where the class sizes are more closely balanced, than test cases C and D. In Figure 5-11, showing average overall performance for each of the five test cases, the test cases A, B and E performed better than C and D when the 54-word set was used, and also overall considering all experiments, although performance in the 65 function words tests is skewed by some poor performance on test cases A.

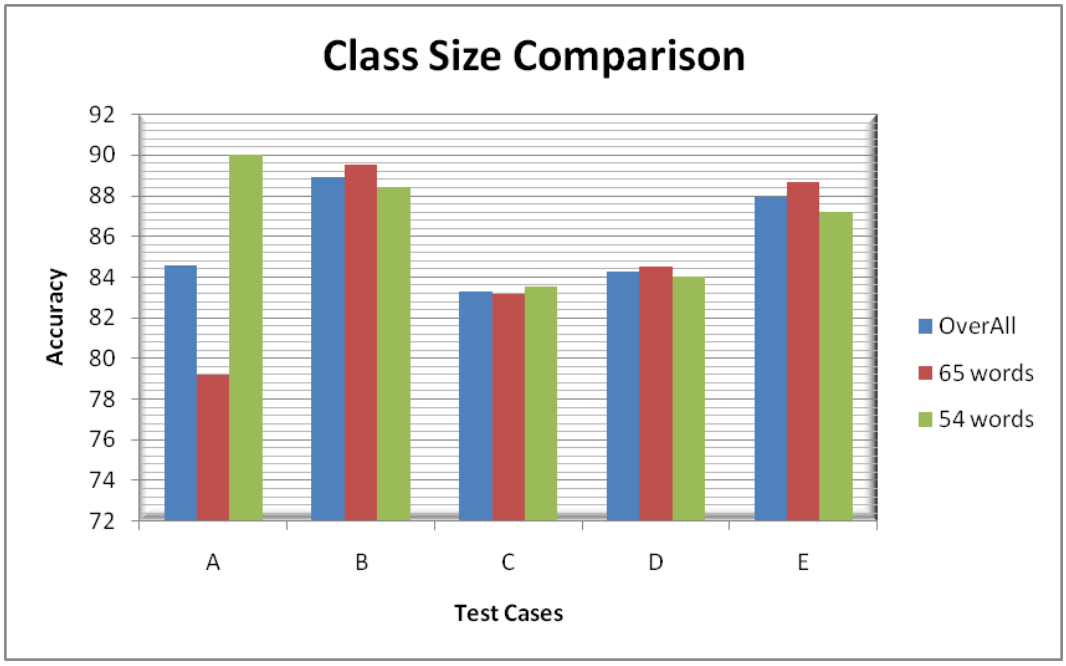


Fig5-11 it compares among the test cases. Every test case has a level of imbalanced classes. A,B and E have slightly imbalanced classes performed better that the more imbalanced cases C and D.

5.6 Summary

The Arabic Authorship attribution problem is a very interesting topic which has not been investigated before. The Arabic Language itself is different in origin and nature from other languages involved in existing authorship attribution research, so it was unclear if similar approaches, especially a function word approach, could be appropriate. Exploring the Arabic language for words that could play the same successful role in authorship attribution as function words in English was a challenge whose chances of success were uncertain when we began this work.

Succeeding in finding a collection of Arabic words that worked as function words is a significant achievement in the Arabic Authorship Attribution field. Extracting this collection of words from the Arabic language came after many researches and experiments. Arabic function words proved that they are good features for extracting authors' fingerprints in a way that serves authorship attribution. Notice that in the experiments we achieve classification accuracies usually well over 80% - but these are accuracies for classifying individual chunks. When we assign authorship to a book, we would use a suitable threshold, perhaps 70%, and say that if more than this threshold of the chunks were attributed to author A, then the book was written by author A. In this sense, we can see that 100% success was achieved in all experiments.

This shows that the Arabic function words satisfy the same role as function words in English, being related to the author's writing manner more than to the text topic, since experiments were done on five quite different test cases. Meanwhile, factors such as chunk size, number of iterations, and the regularization parameter played a role in improving the overall performance, and this would be an important role in individual cases where the overall accuracy would be close to the chosen threshold for attribution.

6

Character Level and Hybrid Feature Representation

Chapter overview

In this chapter we look at alternatives to the pure function-word based approach that was explored in the last two chapters. Section 6.1 describes character level features in general, and section 6.2 describes some experiments on our test data using Arabic Character Level features. Section 6.3 then returns to function-word features only, and looks at the idea of combining function words of the 1000 and 2000 words chunk sizes to form a new set of combined features. Section 6.4 then looks at combining function words and unigram character level features to form another new way of representing a data chunk. Section 6.5 draws some observations and conclusions from the experiments in this chapter.

6.1 Character Level Features

Character-level features are based on the letters or components of the language which combine together to make words. In the authorship attribution context, character level features mean that we use frequencies of characters, or frequencies of combinations of characters, but where the combinations (e.g. pairs of characters) are usually smaller than words. The character-level features that have tended to be used most in research are unigram, bigram and trigram features. In the 'unigram' character-level feature, a feature consists of a single character. The unigram features consist of the alphabet of the language. So, the number of features in the unigram level is the same as the alphabet size. For example, the English alphabet is 26 letters so the number of features in the unigram level would be 26. A bigram feature is based on the frequencies of sequences of two letters. The bigram level consists of a number of features that is the square of the alphabet size. Finally, trigram features are also sometimes explored, which are sequences of three letters. In English, there are 17,576 potential trigram features. Beyond trigram, such as 4-gram, such features are rarely explored because the number of possibilities becomes too large.

Character level features have sometimes been used in authorship attribution studies with success, and there might be reasons to suppose the higher level *n*-grams could be more successful. But when *n* grows to 3, 4 or more, many of the individual features are also words of the language. Sometimes, depending on the topics of the texts under consideration, these words might cause noise and confusion for classification studies.

It is interesting to investigate how much of a difficulty this could be. For English n-grams, I found useful resources for this investigation that are associated with the official Scrabble word game. The rules of the game state that any word must be found in a standard dictionary. There are two sources that produce these lists of acceptable words. The first source is The Official Tournament and Club word list (TWL) which is the official list for Canada, USA and Thailand. The second source is the SOWPODS, which is the official word list for other countries. According to the two resources, three letters could make about 1015 words in TWL and about 1292 words in SOWPODS. At the 4-gram level the number of words are 4030 words according to TWL and 5454 words according to SOWPODS.

These numbers of words are not considered large enough to create confusion in the classification process. Confusion could happen if the number of words is huge because some of these words will be topic-related and therefore misleading to the attribution task. But in the case of the trigram and 4-gram the percentage of words does not exceed 7% of the number of features in the trigram and 1.5% in the 4-gram level. Of course, the more the letters increase is the more the meaning words are found. Table 6-1 shows the TWL numbers of words for different numbers of letters and Table 6-2 shows the same for SOWPODS.

Number of letters	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of words	101	1,015	4,030	8,938	15,788	24,029	29,766	29,150	22,326	16,165	11,417	7,750	5,059	3,157

Table 6-1 shows the number of words that has meaning in different *n*-grams according to TWL. E.g. for bigrams, 101 of them are words with meaning.

Number of letters	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of words	124	1,292	5,454	12,478	22,157	32,909	40,161	40,727	35,529	27,893	20,297	13,857	9,116	5,757

Table 6-2 Shows the number of words that has meaning in different *n*-grams according to SOWPODS. E.g. of all the trigrams, there are 1,292 words with meaning.

Trying to do the same investigation for the Arabic language was unsuccessful because there is no official organization that keeps track of Arabic words as in English.

6.1.1Arabic Character-Level Features

The Arabic alphabet consists of 28 letters. Some of the Arabic letters look similar but are different in sound from one another, indicated by dots above or below the letter. For example, letters numbered 4, 5 and 6 in Table 6-3 are the same, but with different dots that distinguish them. In Table 6-3 there are several examples where letters differ only in terms of the placement of special marks, which change the sound and the pronunciation. The number of possible unigram features is therefore not 28, but it is 35 features. The 35 features are the 28 original letters plus 7 more letters with added diacritics. These are special diacritic signs called 'hamza' ء which added to some letters create different pronunciation and leads to change in meaning. Table 6-3 shows the 35 letters used in the experiments.

1- ا	2- أ	3- إ	4- ب	5- ت
6- ث	7- ج	8- ح	9- خ	10- د
11- ذ	12- ر	13- ز	14- س	15- ش
16- ص	17- ض	18- ط	19- ظ	20- ع
21- غ	22- ف	23- ق	24- ك	25- ل
26- م	27- ن	28- ه	29- و	30- ؤ
31- ئ	32- ى	33- ي	34- ة	35- ء

Table 6-3 shows the Arabic alphabet and the seven letters with added diacritics

The highlighted cells in Table 6-3 are the 7 letters with the diacritic hamza. They are like the original letters but with a different form that changes the pronunciation. For example, the word "search" in Arabic is" أبحث" which starts with the letter "Alef" which is letter number 2, is considered the same starting letter in the word "to", which is "إلى" in Arabic but the placing of the diacritic hamza ء differs, which changes the way the letter is pronounced in the two words. Also, letter number 29 "و " can be also altered by adding hamza to it, as in the two words "صورة" which means picture and "السؤال" which means question. In the last example, the two words have the same letter but the diacritic changes the letter pronunciation according to the word.

6.2 Experiments and Results

Experiments are done with 35 letters which are the original 28 and the 7 modified ones. We use unigram features. That is, every feature contains only one letter, which gives 35 features. The chunks size used in the character-level experiments are different from the sizes used in the function words experiments. The chunks in the character-level cases are not chunks of words but they are letters. Initially I tried to carry out experiments

using the same chunk sizes as in function words (1000, 2000 and 3000), but it turned out that these numbers of letters are too small to reflect writing patterns of the authors. 1000 letters cover about 190 to 200 words, which holds very little stylistic information. So, 5000 letters chunks and 10000 letters chunks are the two main chunk sizes used in these experiments.

The test cases are the same five test cases used in the function words experiments. As before, every text is divided into a number of vectors according to the chunk size. Each vector contains 35 values. Every value is the frequency of one letter among the 5000 or 10000 letters in the chunk.

Table 6-4 shows the number of vectors of the 35 features in each test set for both the 5000 and 10000 chunk size cases. Each test set contains two books representing two authors as explained in chapter 5.

.

	5000 letters		10000 letters	
Test Set	**Author1 Book**	**Author2 Book**	**Author1 Book**	**Author2 Book**
A	12x35	13x35	6x35	6x35
B	17x35	16x35	8x35	8x35
C	36x35	24x35	18x35	12x35
D	26x35	17x35	13x35	8x35
E	16x35	24x35	8x35	12x35

Table 6-4 shows the difference in class sizes in terms of the number of data vectors (chunks) per class (book), when using the unigram character level approach..

Our results from the function word experiments gave us an indication of suitable parameter values to use in the EA/GLM approach. E.g. all the experiments reported here for the character-level features are 300-iteration runs, based on observations from chapter 5, we search 20 values of the regularization parameter.

As before, every experiment is carried out on the five test cases, and ten independent runs of 300 iterations each are done for each test case. The result of every run is a best-performing chromosomes (a subset of features that are to be used as input for a GLM model). Then, a test is carried out using the test books in every case. Test book1 always refers to the book of the first writer and test book2 refers to the second writer. Finally, the average performance of the ten tests is recorded for every case.

6.2.1 Experiments with 5000-character chunks

Table 6-5 shows the results of the 5000 character chunks experiments. The average performance of the five test sets is 80.14%. The best performance is from A and the worst from test case B.

5000 letters and 300 iterations	
Case Name	**Performance**
A	85.88765
B	62.14745
C	88.269
D	88.21565
E	76.1787
Total Mean	**80.13969**

Table 6-5 shows the results of the 5000 letters chunk with 300 iterations.

6.2.2 Experiments with 10000-character chunks

Table 6-6 shows the results of the 10000-character chunk experiments. The best performance again comes from test case A, and the worst is again from test case B. Mean performance is very similar to the previous experiment (5000-character chunks) at 80.11%.

10000 letters and 300 iterations	
Case Name	**Performance**
A	91.21855
B	64.8517
C	82.0886
D	88.9717
E	73.4097
Total Mean	**80.10805**

Table 6-6 shows the results of the 10000 letters chunks with 300 iterations.

Principal component analysis was applied to investigate the weak performance for test case B, and contrasted with test case A. In Figure 6-1 we show a graph of the first two principal components for test case A. Clearly the classes are well separated by the two main components.

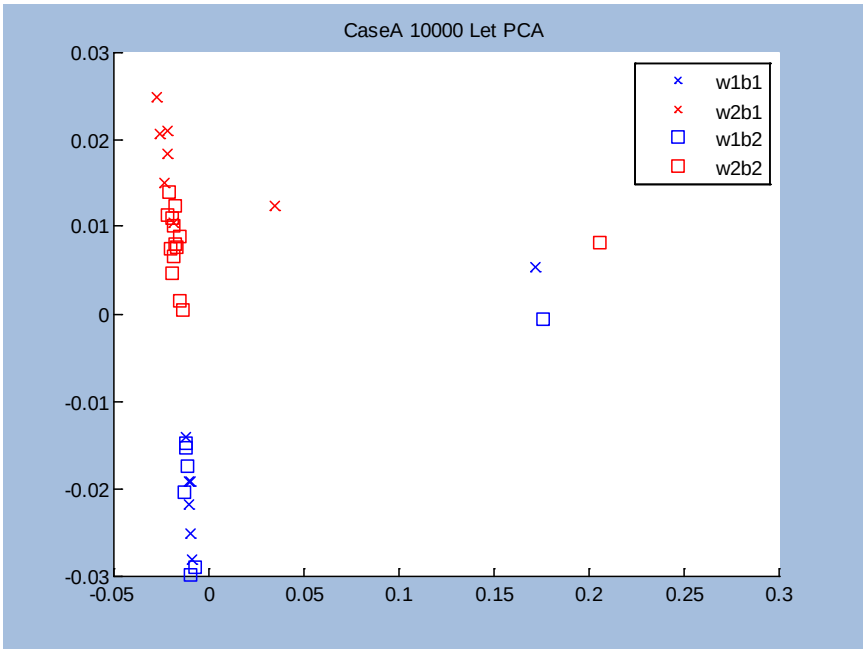


Figure 6-1 shows the Principal Component Analysis of Case A of the Letters in the 10000 letters chunk size. The two sets of authors books are well separated.

On the other hand, Figure 6-2 shows the same plot for test case B. The graph clearly shows that the features of the two authors overlap in way that makes it hard to discriminate between them on the basis of these character-level features. On further analysis we found evidence that in this case it was likely due to the bias in the character frequencies caused by the topics in the books in test case B.

In comparison, Figure 6-3 again shows the two principal components for test case B, but this time at the function-word level, from one of the runs using the 54-word function set and chunk size 2000. The function word features enabled a clear separation between the two authors in stark contrast to the character level situation. In accuracy terms, 54-function words and chunk size 2000 accuracy for test case B reached 89%, compared to 64% for the unigram character level representation.

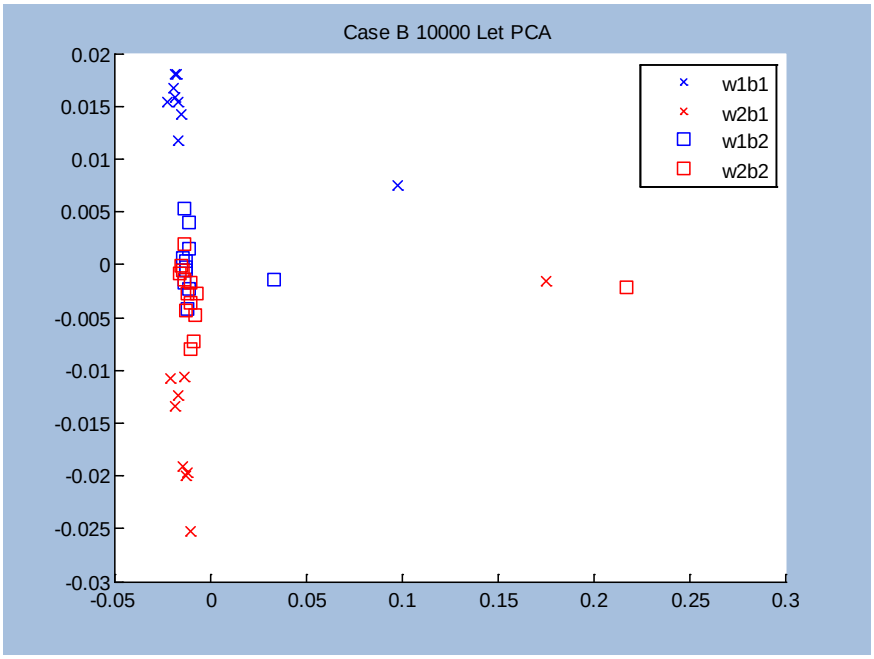


Figure 6-2 Shows the Principal Component Analysis of case B of the 10000 letters chunk size. The two sets of Authors books are highly overlapped**.**

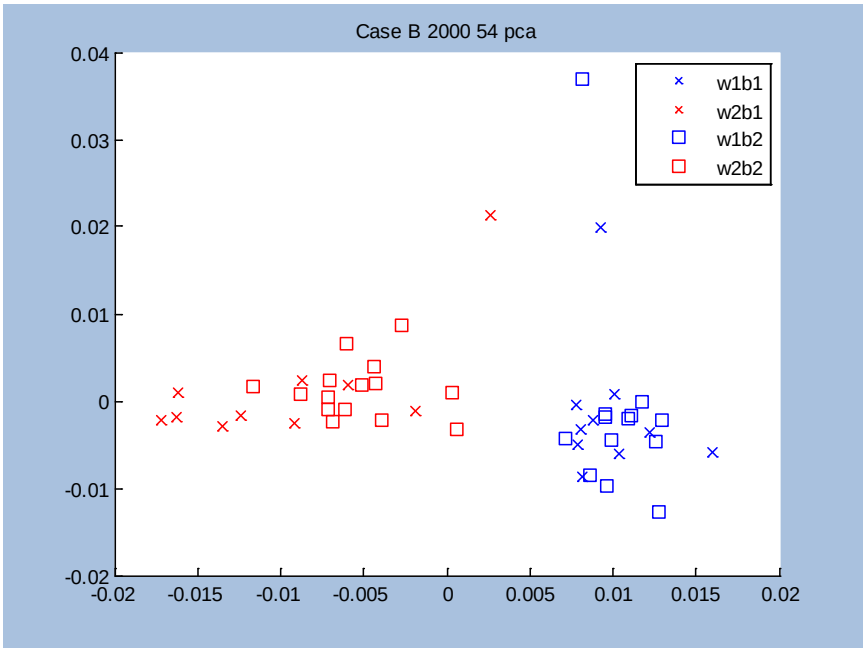


Figure 6-3 It shows the principal component analysis of Case B in function words of the 2000 words chunk size. The two author's books are completely separated.

6.3 Combining Function Words of 1000 and 2000 Words Chunk

As we have discussed before, there is a trade-off to consider between the size of a chunk, and the quality of the information provided for machine learning. When the chunk sizes are small, there is enough data for machine learning methods to have a chance to find models that can discriminate reliably between classes. However, the small chunk sizes mean that each data vector may not capture the author fingerprint very well. With larger chunk sizes, we might expect the author's style to be better reflected in each chunk, but the number of vectors is smaller, and machine learning will be more likely to over fit and be more influenced by noise. However, there is no reason to prevent us from combining smaller chunk sizes and larger chunk sizes. In this section we report about experiments where we combine frequency vectors from 1000 words chunks and 2000 words chunks. The basis of this representation is the 2000 words frequency vectors, based on the set of 54 function words. However each data vector also contains more information, with an extra two sets of 54 frequencies. These are the frequency vectors for the two 1000 word chunks within the 2000 word chunk.

Table 6-7 shows the performance of this combined method. The overall mean value, which is 87.99%, represents the best performance achieved on the test cases in all of the experiments.

54 Function words / cobined 1000 and 2000 words chunk representation	
Case Name	**Performance**
Case A	90.7976
Case B	89.4753
Case C	85.5516
Case D	86.0932
Case E	88.0186
Total Mean	**87.98725**

Table 6-7 shows the performance of combining function words average frequencies of 1000 and 2000 words chunk.

6.4 Combining Function Words and Character Level Features

A further experiment was done to investigate combining the function word approach with the character level approach. We investigated this by using a feature vector of 89 numbers, with the 54 function word frequencies followed by the 35 Arabic unigram characters. We combined the average frequencies of the 54 function words in the 2000

words chunk size, and the 35 unigram characters in the 10000 letters chunk size. In the five cases we compared between the number of chunks of the function words and the letters. We found that there is difference in the number of chunks so we made all the files of equal number of chunks by eliminating the more chunks. For example, if one of the files of function words has 8 chunks of text and the letter file has 7 chunks of text. We eliminate the last chunk of the function words to make them equal. But the difference between the number of chunks were not very big because the average of letters in 2000 words is about 10,00 letters We picked these two features because they performed the best in their categories.

Combined Function words and Character Level	
Case Name	**Performance**
Case A	94.13025
Case B	79.20825
Case C	83.28465
Case D	92.20535
Case E	80.78875
Total Mean	**85.92345**

Table 6-8 shows the performance of combining function words and unigram characters.

Table 6-8 shows the results on the five test cases when using the combination of the 54 function words frequencies and the unigram characters. The overall mean, which is 85.92%, is not as good as when combining the 1000 and 2000 words chunks, although this method did produce the best results found for two of the test cases, A and D.

6.5 Conclusion and Observation

The character-level unigram features are able to distinguish between authors, but not as well as the Arabic function words based features. For some of the test cases, such as test case A, the performance with unigram features was particularly good. This suggests that unigram features can be successful in capturing parts of an authors' style fingerprint, however it is not clear whether or not such cases were affected by fortunate biases caused by the content of the books in those cases. Generally, performance with the unigram features was moderate or low, especially with test cases B and E, in both the 5000 and 1000 character experiments.

The idea of combining function word frequencies from both 1000 and 2000 words chunks in the same data vectors led to the best overall mean result in all experiments, while at the same time not providing the best result for any of the individual test cases. This seems to be quite promising for the combined method. It suggests it is quite robust to identifying general author fingerprints.

Table 6-9 shows the overall mean accuracy performances from all of the different types of experiments. The experiments are ranked in terms of accuracy, with the highest appearing at the top of the table. As we have indicated, the best mean performance was achieved by the approach that combined 54 function words (FW) frequencies from 1000 and 2000 words chunks. In second place comes the 54 FW 2000 words chunk size, with 300 iterations. The worst performance comes from the 10000 character unigram frequencies. The best performance for both test cases A and D was achieved in the combined 54 FW and unigram characters experiment. The best performances for test cases C and E were also from experiments involving the 54 FW set, but not in a combination approach, and with chunk sizes of 3000 words and 2000 words respectively. For test case B we find that the best performance was achieved in the 65 FW / 1000 words chunk size experiment.

Rank	Run type	Performance
1	Combined 54 FW of 1000 and 2000	87.99%
2	54 FW of 2000 words 300 iterations	87.67%
3	54 FW of 2000 words 100 iterations	87.25%
4	54 FW of 3000 words 300 iterations	86.56%
5	65 FW of 2000 words 300 iterations	86.46%
6	Combined 54 FW and Letters 300 iterations	85.92%
7	65 FW of 1000 words 300 iterations	84.53%
8	65 FW of 1000 words 100 iterations	84.26%
9	54 FW of 1000 words 300 iterations	84.25%
10	65 FW of 3000 words 300 iterations	83.99%
11	5000 characters 300 iterations	80.14%
12	10000 characters 300 iterations	80.11%

Table 6-9 shows all over performances of all the experiments.

It is clear that using as features the frequencies from the 54 function words set generally led to the highest levels of performance overall. Apart from this it also seems clear that using frequencies from different chunk sizes in the same vector is also very promising. This helps to capture the useful information that can be found from

averaging over more words, but while still keeping the benefit of a large number of data vectors.

7

Conclusion

Chapter overview

In this chapter we present conclusion of the thesis. Moreover we present the summary and the contribution of the research that presented in the thesis. In section 7.1 is the summary of the thesis. In section 7.2 is the contribution of the research and finally section 7.3 is the future work.

7.1 Summary

Authorship attribution is a research field that covers many different kinds of application and research question. Battles over the authorship of one specific text, discovering the author of an anonymous work, and suspicion over the originality of a text are three of the main types of issue that are addressed by authorship attribution studies. Whatever the type of issue that is being addressed, there are also many obstacles that could be encountered, making it difficult to answer the question at hand. Some such obstacles include alterations that may have been made to the original text, the stealing or borrowing of ideas between authors, and also the natural development and variation in a single authors' writing style over time. All such issues can complicate and obscure the process of finding of the true author of a work.

There is no doubt that every author, writer, novelist and any one works in the field of composing text, has his or her own pattern of writing, or 'fingerprint' which can be found in their work. It is generally expected that this pattern or style fingerprint is reflected in text-based features. Discovering the best features to work with is essential to achieve acceptable and desirable results in authorship attribution. There are many different types of features that have been used in authorship attribution research, such as vocabulary, punctuation, words, word length, phrases, sentences and sentence length. Every one of these features has been proposed to help identify some or all of an author's stylistic fingerprint. The amount of work that has been done and the range of different types of features that have been studied reflect the importance of finding the right choice of features for this task.

Function words have proven to be a very popular and successful source of features in English language authorship attribution and related studies. It has been show in many cases that function words can expose and extract style and pattern fingerprints that are specific to individual authors. In this research on English language authorship attribution, as described in chapter 3, function words played a great role in identifying and differentiating between authors in the cases of the Federalist Papers and the 15th book of Oz.

Authorship Attribution research is also carried out in the context of many other languages apart from just the English language. But the amount of research done in the English language context is far greater than in any other language, due to the availability of electronic materials and also the existence of several famous disputed cases to work on. Nevertheless, some research has been done on authorship attribution

in other languages; there are studies relating to Greek, Chinese, Spanish, Dutch, Latin and Croatian, as mentioned in chapter 2.

Understanding the nature, differences and similarities among the origins of languages has helped in understanding how far a feature in one language could be used directly or indirectly for the same task in other languages. As mentioned in chapter 4, languages are related by family trees. So, languages that share the same branch of a family tree tend to have common linguistic features such as grammar and vocabulary. It is clear, as we saw in chapter 4, that English and Arabic are quite far separated, in different language families, therefore we can expect that the idea of using Arabic function words is not straightforward – that is, we could not expect that direct translation from the English function words to Arabic equivalents would be successful. I found this naturally to be the case, from my own knowledge of Arabic and English. Most of the English function words when translated did not match the same usage and role in Arabic as in English. Also, some of the English function words give the same meaning when translated. However, the basic idea of function words – words that play supporting roles in text, that are not related to content – remains an attractive and sensible idea for authorship attribution, and one that I believed would be possible since words exist in Arabic that have such supporting roles. So, I set about creating an original list of Arabic function words. The initial list consisted of 106 words. This list was reduced to 65 as a result of omitting words that were seldom used the texts in our dataset. Finally, we have established that a list of 54 words generally gave rise to high accuracy levels in predicting the correct author in our test cases. The lists of Arabic function words are presented in detail in chapter 5.

Authorship attribution in the Arabic language is a new area of research, but one that is very rich in challenges and opportunities. The literature of the Arabic language is very rich and provides many cases of authorship attribution interest. However there needs to be more interest and activity in converting existing Arabic literature materials into electronic text form, and developing easily used tools for processing these texts, before most of these opportunities can be realised. Such greater availability of electronic texts in Arabic will be a huge benefit for researchers – they will be relieved of the burden that has taken up much of the time during the PhD research, which was concerned with finding and processing Arabic materials and texts to be suitable for experiments and research.

7.2 Contributions

The main contributions of the research presented in this thesis are, very briefly, the first authorship attribution studies in the Arabic language, the discovery of successful features for Arabic authorship attribution, the hybrid EA/GLM algorithm we used in most of the experiments, and finally the dataset and the test cases we prepared, that are available for others to use for further similar studies in Authorship attribution in the Arabic language.

I would like to stress the originality of the topic of this thesis. The research and experiments that are reported in this thesis are the first in the field of Arabic authorship attribution. One of the main achievements of the thesis is to demonstrate the possibility of differentiating between two authors in the Arabic language, using the hybrid algorithm and the proposed features. Moreover, the generation of a successful list of Arabic function words, which helped in extracting the different writing patterns of Arabic authors, is potentially very useful for future research in this field. Also, the significant number of edited and processed datasets used in the research is ready to be used in other research.

The research done previously in English language and the literature review helped me in selecting appropriate tools and features to investigate in experiments for the Arabic language. Although, as discussed, the differences between the Arabic and English languages meant that the generation of possible function words in Arabic was much more than a matter of translation – for this I needed my own knowledge of Arabic, and the gradual process that has been described above.

The hybrid algorithm used in the experiments, presented in details in chapter 5, was shown successful and suitable for the authorship attribution task. The hybrid algorithm finds a good subset of features that seem the best for distinguishing between two authors (when that subset is used as the basis of a GLM) on the training data. The use of a feature selection approach that emphasises small subsets was helpful to avoid over-fitting. However there is no claim here that the EA/GLM approach is particularly strong for authorship attribution. Many other machine learning approaches could be tried, and many appear in authorship attribution studies. It was not the focus of this thesis to develop and evaluate machine learning approaches for this purpose; instead it was recognised that the choice of features to use is the main aspect in such studies (which is the same in some of application domains of machine learning).

Following earlier experiments to develop and test the EA/GLM hybrid algorithm on English language authorship attribution problems, and then the creation and testing of candidate Arabic function word sets, three main groups of experiments were then done in the course of this thesis research: these were (i) the function words group of experiments; (ii) the character level group of experiments; (iii) the combination of features group of experiments.

In the first group of experiments, tests were done for chunk sizes of1000, 2000 and 3000 words. A chunk of text was represented by a data vector that consisted of the frequencies in that chunk of each of the 65, or the 54, function words used in the experiment. Although the results of paired T-tests were sometimes inconclusive, the performance of the 54 function words group was the most promising, and certainly was sufficient to identify the correct author of the book in all test cases. Also, a chunk size of 2000 words was generally the most promising of the chunk sizes.

In the character level experiments, the experiments were divided into two main groups according to the characters chunk size, 5000 and 10,000. These groups of experiments gave the least promising overall results. It is likely that this is due to the fact that the character level features can be strongly influenced by the topic of the text, and this effect introduces noise that obscures the author's style fingerprint.

In the third group of experiments, various combinations of features were tried. When combining function word and unigram character level features, the overall average result was better than the result of the character level only experiments. Moreover, this combination led to the best mean results for the two test cases A and D. However, the character level features are based on all of the words and can therefore, as we say above, be influenced by the subject and content of the work (unlike with function words). Although combining the two features seemed to help in extracting and exposing more of the pattern and fingerprints of the authors, but it is not clear how much this might have been influenced by fortunate biases in the subjects of the test cases A and D.

A different type of combination that we investigated in the third group of experiments was to combine the 54-function words frequency vectors for 1000-word and 2000-word chunks. This achieved the best overall results from all experiments.

A brief summary of the contributions in this thesis is as follows:

- C1: Identification of sets of function words for the Arabic Language, that are successful in the Authorship Attribution problem Studies. The identification and

use of function words in Arabic is original and can form the basis for very much continued work in Arabic text processing.

- C2: The introduction the hybrid (Evolutionary Algorithm / Linear Discriminant Analysis) algorithm, which is shown to be successful for the Authorship Attribution problem in both English and Arabic; in particular, the algorithm finds small subsets of features that are able to form the basis for a GLM classifier. The small size of the subset is considered to be beneficial for generalisation performance, and in its own right this small subset can be useful for further stylistic analysis.

- C3: We present the first investigation and assessment of character level features for the Authorship Attribution problem in the Arabic Language.

- C4: We contribute a significant pre-processed dataset, along with a set of pairs-of-authors test cases that are ready to be used by other studies in the field of Authorship Attribution in the Arabic Language. The datasets are edited and available on request.

- C5: We contribute the ideas of combining function word frequencies from different chunk sizes, as well as the idea of combining function word and character level features. Both provide promising results, especially the former, that suggest these ideas are worthy of further study.

7.3 Future work

Future work in the Arabic authorship attribution field is wide open to an immense number of possibilities because this is a new area of research. An obvious area for further work is the investigation of alternative classification tools. This has not been a focus of this thesis, since we have focussed on the investigation of Arabic function words, suitable datasets, and the appropriate choice of features or feature combinations. We argue that the establishment, in this way, of appropriate features is more important since it will benefit most classification methods. This means that the EA/GLM algorithm has not been fully evaluated by comparison with other methods in the context of Arabic data, however in the context of English authorship attribution problems it was

found that the results achieved were in general comparable or better than all of the published results for the same problems.

One type of alternative classification method of particular interest of further study would be to use alternative classification methods within the EA/GLM approach – that is, perhaps EA/SVM or EA/KNN. This type of combination has been used successfully in many other domains, and will retain the benefits of finding small subsets of features.

An important finding is that 2000 words were most successful as a chunk size. Obviously we would expect this to be different for different sets of authors. However, since our dataset involves many distinct pairs of authors, the experiments have made it clear that such a large amount of words (perhaps equivalent to 3 or 4 pages) is ideal for establishing an author's fingerprint in terms of function word frequencies. This suggests that smaller scraps of text will be much more challenging for author attribution, and that we need additional insights and ideas to classify these. This is significant, because one exciting potential project in Arabic Authorship Attribution is related to small numbers of words. This project is about analysing the quotes that are generally attributed to the prophet Mohamed, and categorizing these quotes according to credibility and authenticity.

There are several lines of further work that could be done to make Arabic authorship attribution possible for smaller scraps of text. Further testing and analysis of the 54-word function set could show us that, for some of the function words, their frequencies in smaller chunks (e.g. 100 words) might be sufficient, and for other words the frequencies might need to be taken over larger chunks. It might be possible to then combine character level features with specially selected function words to provide an approach for smaller scraps of text.

Finally, a deeper analysis will be interesting to do of the small subsets of function words that are found by the EA/GLM method. If we look at the results of many test cases, we might find that some words are selected more than others, and these can therefore be given greater weights in a changed version of the EA/GLM algorithm. However we think that one good future direction is to investigate how the prominent function words like these are used together in sentences. This provides other features, such as frequency of word A appearing before word B in a sentence. This type of feature may also be helpful for the problem of authorship attribution with smaller scraps of text.

Appendix A

Experiments results

Detailed results of all the experiments in order as presented in chapters 5 and 6.

Chapter 5: Arabic function words.

1- Experiments' results for 65 words and 100 iterations of the1000 words chunk

Case A

Run	Writer	Writer2
1	72.6190	61.0632
2	70.3297	80.5419
3	92.8571	83.9080
4	80.9524	69.5402
5	58.5714	65.7825
6	68.4211	74.4102
7	86.1607	88.0637
8	92.3469	87.9310
9	66.3265	87.2679
10	88.5714	75.4991

Case B

Run	Writer1	Writer2
1	94.5238	92.5926
2	81.5972	93.9445
3	90.9722	68.2759
4	93.1818	82.2281
5	92.9825	99.0805
6	98.8971	82.3448
7	84.0278	93.1034
8	89.0805	99.6865
9	91.7793	95.0000
10	95.8333	76.8025

Case c

Run	Writer1	Writer2
1	86.9565	70.9259
2	79.5396	53.8462
3	79.4050	93.8131
4	88.1007	78.9474
5	95.3416	42.8571
6	81.4010	90.6863
7	75.6522	59.3434
8	78.5326	78.2118
9	98.0435	75.6173
10	82.4534	90.1786

Case D

Run	Writer1	Writer2
1	72.8745	84.2105
2	90.6883	81.5789
3	63.9098	96.4912
4	82.1510	94.9561
5	99.0431	95.3947
6	71.0526	81.5789
7	89.2632	94.1520
8	78.0186	85.4489
9	93.5673	95.6938
10	92.6316	72.9323

Case E

Run	Writer1	Writer2
1	96.7033	96.8815
2	91.6667	88.6978
3	79.1925	80.0515
4	87.9870	80.8237
5	92.8571	84.2905
6	93.2331	99.8578
7	95.6522	73.4878
8	86.2857	74.5946
9	93.6090	94.5946
10	93.6090	98.1508

2- Experiments' results for 65 words and 300 iterations of 1000 words chunk

Case A

Run	Writer1	Writer2
1	70.7231	69.2084
2	82.8407	49.0749
3	67.9402	72.7586
4	68.3673	64.7676
5	98.3389	87.5254
6	98.3871	79.7729
7	92.0000	76.0188
8	99.7321	58.9080
9	63.6975	71.2047
10	67.3030	83.5215

Case B

Run	Writer1	Writer2
1	94.3034	64.8276
2	91.1404	97.5830
3	95.8333	65.1341
4	81.1469	98.9568
5	88.9798	94.5537
6	89.3981	96.8688
7	93.7500	89.2414
8	98.8956	91.3237
9	90.0862	97.4394
10	97.2222	85.6401

Case C

Run	Writer1	Writer2
1	91.4240	81.2028
2	83.3623	64.4841
3	90.5877	39.2857
4	90.1943	55.4074
5	95.6932	71.0678
6	77.1429	82.0195
7	95.6856	86.7187
8	79.0082	77.0602
9	96.3067	82.9861
10	88.0182	76.0913

Case D

Run	Writer1	Writer2
1	97.7172	89.9342
2	88.7218	80.8612
3	77.2311	90.5702
4	88.7135	98.2784
5	65.4762	71.6374
6	88.1223	94.4120
7	91.3269	59.5395
8	86.5841	89.5243
9	95.6050	82.6316
10	98.9906	92.8623

Case E

Run	Writer1	Writer2
1	89.1865	85.4681
2	86.7596	65.1106
3	100.0000	91.0511
4	92.6136	91.2995
5	84.7176	86.5946
6	93.5065	88.5040
7	94.5853	94.8949
8	94.5714	95.1014
9	89.1741	95.8193
10	87.6026	89.7075

3- Experiments' results for 54 words and 300 iterations of 1000 words chunk

Case A

Run	Writer1	Writer2
1	94.2857	100.0000
2	93.8776	100.0000
3	62.3153	100.0000
4	74.2105	100.0000
5	77.4160	100.0000
6	47.2085	100.0000
7	83.7535	100.0000
8	94.5215	100.0000
9	49.8496	100.0000
10	74.7253	100.0000

Case B

Run	Writer1	Writer2
1	99.1519	80.9044
2	92.0412	96.9596
3	94.3681	76.1755
4	86.5546	98.3990
5	91.8317	97.5327
6	96.4888	97.5096
7	94.6385	67.8826
8	92.2893	84.6327
9	89.4520	97.5745
10	94.1327	81.3588

Case C

Run	Writer1	Writer2
1	86.1082	85.8025
2	79.3065	81.6817
3	91.1418	66.8780
4	88.8696	73.3572
5	94.2768	68.8630
6	99.0809	37.3611
7	78.3057	85.1667
8	69.8849	85.5287
9	93.9887	63.5696
10	90.4527	56.9946

Case D

Run	Writer1	Writer2
1	98.2718	78.1445
2	85.3411	97.0803
3	95.4887	88.0000
4	62.3904	78.1798
5	99.6491	87.5515
6	94.0351	75.0424
7	98.5380	77.2701
8	79.5268	88.6375
9	95.6491	88.6346
10	90.0405	82.1220

Case E

Run	Writer1	Writer2
1	87.9840	81.3990
2	88.9163	76.2677
3	94.4251	66.8117
4	82.0728	92.4538
5	81.1688	71.8350
6	93.4608	90.2076
7	88.5714	82.4324
8	95.7792	92.6263
9	94.2716	82.3906
10	82.0513	66.9630

4- Experiments results for 54 words and 100 iterations of 2000 words chunk using 10 alphas values

Case A

Run	Writer1	Writer2
1	94.9451	89.8437
2	74.5174	75.0965
3	98.7395	97.6959
4	99.6678	96.3265
5	99.5074	97.7833
6	58.2210	69.2722
7	66.9388	54.8872
8	97.2527	45.0549
9	74.4898	62.9630
10	88.0000	88.8571

Case B

Run	Writer1	Writer2
1	91.4634	95.5150
2	91.5033	87.7976
3	97.6776	57.6112
4	92.0635	95.9416
5	89.7436	91.6883
6	89.2276	98.3965
7	89.4097	94.5055
8	93.0000	95.8571
9	94.7368	60.9127
10	97.0339	80.3571

Case C

Run	Writer1	Writer2
1	86.2170	85.5556
2	89.4428	85.1254
3	82.8877	85.9053
4	81.5851	73.0640
5	93.8636	48.0392
6	88.1423	83.0484
7	89.3506	78.8530
8	89.4349	72.2222
9	93.2900	77.4306
10	92.0000	63.8889

Case D

Run	Writer1	Writer2
1	93.0931	94.1520
2	99.0338	87.4396
3	76.8519	84.0000
4	95.4733	52.3148
5	93.7669	73.1707
6	94.6667	99.5726
7	95.4416	54.3544
8	97.3856	98.0952
9	99.2908	95.5556
10	96.2963	52.0833

Case E

Run	Writer1	Writer2
1	95.6710	90.1042
2	95.5556	87.5604
3	94.2085	89.9691
4	81.1847	94.4444
5	90.2439	92.4119
6	82.0276	83.5017
7	92.1875	87.0674
8	95.5357	93.5484
9	91.7460	84.0909
10	86.3636	82.2751

5- Experiments' results for 54 words and 100 iterations of 2000 words chunk of 20 alpha values

Case A

Run	Writer1	Writer2
1	77.9221	77.8571
2	100.0000	93.2432
3	98.6711	95.3488
4	53.0075	58.7302
5	91.4966	92.8571
6	93.4066	81.3492
7	99.3789	94.1392
8	100.0000	93.4066
9	84.8980	91.3420
10	97.6190	89.7059

Case B

Run	Writer1	Writer2
1	89.5425	95.8791
2	90.3986	98.2143
3	92.6829	75.2613
4	91.6667	92.1927
5	90.7609	82.7988
6	92.2619	70.9302
7	87.8205	92.6871
8	83.0357	74.0047
9	95.0000	75.7764
10	90.6504	97.6190

Case C

Run	Writer1	Writer2
1	98.6226	78.0303
2	90.0000	83.3333
3	84.5209	84.2342
4	96.1877	74.1667
5	92.1630	70.6349
6	77.1499	89.2157
7	91.6084	80.3419
8	90.2357	62.2685
9	89.7361	85.1852
10	57.4545	77.7778

Case D

Run	Writer1	Writer2
1	85.3801	70.9877
2	93.2099	57.3099
3	94.2029	91.9192
4	86.8421	54.5752
5	95.5556	98.8889
6	95.6072	90.4321
7	92.2222	89.2720
8	90.2778	95.6229
9	100.0000	90.9722
10	93.4783	94.9074

Case E

Run	Writer1	Writer2
1	91.1111	90.9877
2	91.5058	91.7417
3	89.2063	92.5505
4	96.1039	84.4444
5	93.5310	87.0370
6	93.5065	85.5556
7	83.5404	90.8983
8	82.7731	88.8889
9	89.5238	88.7597
10	93.0502	83.0484

6- Experiments' for 54 function words and 300 iterations of 2000 words chunk using 10 Alpha values

Case A

Run	Writer1	Writer2
1	99.4048	95.0180
2	95.1247	96.2834
3	94.0199	91.9126
4	90.3262	80.6574
5	62.4424	64.1604
6	66.8405	69.9019
7	96.2348	91.6345
8	93.8492	88.4793
9	85.8466	86.5229
10	96.8944	92.5770

Case B

Run	Writer1	Writer2
1	91.9978	89.5640
2	91.6076	84.9896
3	92.0213	84.9315
4	87.6590	85.5042
5	90.5329	82.6598
6	89.3519	84.4203
7	91.0920	96.5517
8	90.5685	91.5016
9	89.9876	93.5020
10	90.4672	90.3941

Case C

Run	Writer1	Writer2
1	92.6962	59.9010
2	81.3692	80.1783
3	90.4959	74.8818
4	98.2720	72.1100
5	96.4427	63.4259
6	93.3155	76.5251
7	89.8046	76.8254
8	94.9060	82.5843
9	78.8656	84.0708
10	97.0219	73.0496

Case D

Run	Writer1	Writer2
1	99.5885	56.5657
2	96.9697	92.1017
3	66.7944	57.3856
4	89.6104	89.0228
5	84.1751	96.4052
6	97.9391	87.4883
7	85.6102	91.9831
8	98.9691	69.4444
9	85.7143	63.0303
10	93.0556	91.7001

Case E

Run	Writer1	Writer2
1	90.3106	89.7585
2	81.2760	89.7059
3	90.1202	86.1371
4	90.8425	83.7838
5	84.2986	91.0218
6	94.4048	85.4589
7	90.7563	90.7778
8	91.7826	88.6431
9	88.2427	86.2863
10	86.4407	89.5719

7- Experiments' Results for 54 function words and 300 iterations of 2000 words chunk using 20 Alpha values

Case A

Run	Writer1	Writer2
1	98.2402	93.3673
2	98.9876	92.5493
3	99.0033	94.8897
4	99.8829	96.5561
5	91.5888	92.9286
6	95.3917	92.1429
7	98.5931	94.7497
8	93.5714	90.7910
9	97.6546	64.5542
10	96.9466	93.6224

Case B

Run	Writer1	Writer2
1	88.0051	95.6947
2	90.1042	92.1714
3	85.4308	81.7308
4	89.0152	88.6555
5	90.1515	92.9054
6	90.9247	78.3644
7	92.9798	78.1696
8	87.0711	70.3722
9	85.8122	79.1244
10	90.0997	78.6935

Case C

Run	Writer1	Writer2
1	81.0836	86.4734
2	95.3125	71.2073
3	79.5455	90.6463
4	81.6043	72.4900
5	93.4426	78.0501
6	95.0089	69.9074
7	96.8219	66.0819
8	83.3085	86.9308
9	83.0303	75.6614
10	90.5095	79.9197

Case D

Run	Writer1	Writer2
1	98.4127	89.0598
2	98.2323	78.8889
3	60.9877	96.7251
4	95.8995	76.2873
5	94.1043	81.6010
6	90.5070	81.1111
7	89.5921	66.1765
8	86.8120	68.8209
9	95.8848	65.7005
10	94.2029	95.2719

Case E

Run	Writer1	Writer2
1	89.7959	91.4849
2	91.8012	87.5876
3	93.1732	92.8493
4	90.6122	88.8889
5	91.2621	89.4822
6	86.5546	86.4198
7	91.9137	88.4444
8	91.5528	88.1313
9	92.4964	90.0000
10	91.4553	90.2222

8- Experiments' results for 65 function words and 100 iterations of 2000 words chunk using 10 alphas values

Case A

Run	Writer1	Writer2
1	100.0000	93.3673
2	77.2321	79.0476
3	100.0000	64.0977
4	84.7826	89.3688
5	100.0000	95.9707
6	86.1472	94.5238
7	81.3187	90.5714
8	94.5378	95.9524
9	79.7619	71.6270
10	84.3091	87.1429

Case B

Run	Writer1	Writer2
1	88.9881	86.7347
2	91.0131	93.2857
3	91.0714	72.2997
4	90.4762	73.7327
5	87.3932	94.8980
6	90.9574	74.1883
7	87.3656	86.8726
8	97.5309	74.7354
9	91.8651	76.7007
10	85.1010	95.6767

Case C

Run	Writer1	Writer2
1	95.9893	38.0658
2	91.4894	81.2169
3	86.5455	44.6860
4	72.7273	92.3611
5	79.6537	88.3041
6	93.7799	57.1895
7	95.1220	60.4762
8	87.1355	81.9172
9	96.7742	67.0635
10	95.4545	71.9219

Case D

Run	Writer1	Writer2
1	86.3799	96.7742
2	96.4646	93.4343
3	100.0000	33.6601
4	98.7302	75.1852
5	96.9349	83.5391
6	99.1667	73.0159
7	65.4971	99.5169
8	95.8333	94.4444
9	80.7256	64.3939
10	100.0000	64.8746

Case E

Run	Writer1	Writer2
1	89.7010	85.4167
2	88.6364	89.3519
3	78.3550	89.8990
4	90.4762	87.0915
5	91.5751	86.9444
6	90.1099	86.7284
7	82.7922	80.7494
8	89.6104	85.7843
9	90.4762	85.1111
10	92.6829	85.7550

9- Experiments' results for 65 function words and 300 iteration of 2000 words chunk using 10 alphas values.

Case A

Run	Writer1	Writer2
1	99.0185	93.3036
2	97.8836	95.9016
3	99.6849	93.7343
4	63.2094	72.4049
5	98.6216	91.0000
6	78.9916	69.6241
7	96.9866	94.6595
8	85.2571	73.7685
9	97.9929	92.3024
10	77.1429	83.2451

Case B

Run	Writer1	Writer2
1	89.0457	85.7697
2	89.5317	91.3143
3	90.2412	97.4194
4	91.3832	91.5851
5	87.7193	97.6517
6	89.4444	84.7015
7	87.1528	96.8695
8	94.0289	79.5055
9	82.9457	84.6783
10	90.7475	88.0773

Case C

Run	Writer1	Writer2
1	97.9887	69.2354
2	83.9949	88.2177
3	98.1061	65.2237
4	96.7965	78.3730
5	87.6680	52.5926
6	95.2234	74.4244
7	81.3747	86.3512
8	76.8526	83.7798
9	81.7087	80.1282
10	85.7294	80.1405

Case D

Run	Writer1	Writer2
1	96.1948	91.1111
2	84.5833	96.7320
3	91.7929	94.7090
4	88.2963	74.8858
5	61.4540	83.8019
6	84.4444	93.7500
7	68.9519	88.6541
8	98.2385	56.9444
9	89.6552	61.3248
10	95.8606	89.9676

Case E

Run	Writer1	Writer2
1	91.9378	85.8092
2	91.5584	84.4237
3	88.3459	85.8025
4	93.7318	86.9048
5	92.3899	84.2415
6	89.8571	88.6621
7	88.8199	85.3030
8	94.0092	87.7653
9	88.8571	79.6656
10	91.4928	88.0392

10- Experiments' results for 54 function words and 300 iterations of 3000 words chunk

Case A

Run	Writer1	Writer2
1	84.7973	100.0000
2	67.6829	100.0000
3	70.6349	100.0000
4	93.8776	100.0000
5	81.1594	100.0000
6	82.2848	100.0000
7	79.8986	100.0000
8	80.9524	100.0000
9	69.6154	100.0000
10	60.7407	100.0000

Case B

Run	Writer1	Writer2
1	94.2308	82.5847
2	94.3627	84.5926
3	93.9716	82.8704
4	93.3712	80.6397
5	94.0000	79.5556
6	91.7593	79.3651
7	95.1786	84.8413
8	95.2381	81.4307
9	96.2591	85.6551
10	94.4268	84.7793

Case C

Run	Writer1	Writer2
1	96.3526	81.4338
2	96.2693	85.5516
3	96.7033	85.5903
4	93.9901	80.8685
5	89.8268	64.0027
6	97.9853	85.5960
7	97.8162	87.9310
8	97.1014	80.3922
9	96.7885	84.3664
10	93.3405	88.5013

Case D

Run	Writer1	Writer2
1	92.0068	71.0069
2	99.0741	70.1320
3	89.4081	63.1410
4	89.9340	73.7548
5	82.0402	77.5000
6	95.6140	78.0914
7	97.0370	86.4428
8	75.0000	65.1815
9	52.0325	91.9298
10	82.1522	65.7258

Case E

Run	Writer1	Writer2
1	82.1691	93.6275
2	62.5926	91.6667
3	79.8438	92.9570
4	88.7987	94.6391
5	78.5461	93.8380
6	78.1022	94.0647
7	90.9615	92.8205
8	72.7273	95.6573
9	79.5276	94.4667
10	88.9423	91.2854

11- Experiments' results of 65 function words and 300 iteration of 3000 words chunk

Case A

Run	Writer1	Writer2
1	78.2374	72.1417
2	77.1429	61.7552
3	89.5270	69.1441
4	76.0081	72.7599
5	86.8217	61.7284
6	96.9937	62.9395
7	73.1061	64.4272
8	77.7985	69.1542
9	68.8849	68.1055
10	75.0000	72.9901

Case B

Run	Writer1	Writer2
1	95.1199	84.7510
2	90.4959	91.7760
3	95.4291	82.0943
4	93.4821	70.6635
5	94.1434	79.4929
6	97.2711	80.1641
7	96.5328	80.1471
8	91.1526	84.8418
9	97.4662	90.2963
10	95.4082	84.5977

Case C

Run	Writer1	Writer2
1	95.4082	78.1630
2	95.6116	82.3129
3	88.0423	67.0543
4	96.9452	82.1581
5	97.6190	79.0073
6	94.0789	74.6981
7	97.6355	84.7015
8	95.2381	85.3009
9	91.2801	78.0303
10	96.7770	82.9618

Case D

Run	Writer1	Writer2
1	88.4956	77.6316
2	81.3390	83.6391
3	99.7076	61.5265
4	80.7443	80.0314
5	100.0000	74.3802
6	92.9293	72.8464
7	99.1667	73.5876
8	96.9565	84.5679
9	96.3836	66.8333
10	99.2424	39.2720

Case E

Run	Writer1	Writer2
1	86.2500	91.3768
2	90.6716	88.1043
3	90.7895	90.0763
4	85.4545	89.7727
5	76.9802	91.8367
6	78.3482	89.1437
7	86.8952	91.9377
8	84.1102	92.2969
9	86.6803	94.1176
10	84.6154	86.7560

Chapter 6: Character level and hybrid feature representation

12- Experiments' results of 5000 characters- chunk

Case A

Run	Writer1	Writer2
1	90.7018	80.3043
2	87.7148	83.0000
3	88.5739	79.7500
4	94.1606	82.5455
5	81.6092	82.5051
6	91.7460	87.5269
7	89.3204	80.4660
8	88.9184	85.7662
9	94.0883	84.1739
10	85.9929	78.8889

Case B

Run	Writer1	Writer2
1	70.8929	68.0873
2	71.6002	62.5175
3	92.8833	72.3277
4	48.3292	41.5018
5	78.2704	64.2713
6	40.4233	66.4497
7	91.6763	67.5641
8	21.1355	65.1726
9	25.3830	32.4671
10	90.3475	71.6484

Case C

Run	Writer1	Writer2
1	78.4639	98.5176
2	81.3677	97.6024
3	80.9943	98.7674
4	79.6970	97.1726
5	77.4704	97.7618
6	79.7748	98.8130
7	82.4627	94.8798
8	81.4359	99.0546
9	69.8276	92.2437
10	80.9524	98.1203

Case D

Run	Writer1	Writer2
1	75.1303	77.9454
2	94.1176	89.7270
3	96.8031	92.0709
4	93.7140	90.7366
5	99.8529	98.8924
6	94.2199	75.2976
7	63.3910	80.2989
8	85.7505	93.2609
9	79.7226	93.8508
10	91.4099	98.1203

Case E

Run	Writer1	Writer2
1	85.8543	87.4873
2	40.9836	51.0721
3	62.1069	70.3976
4	80.8333	87.7778
5	55.1347	78.4937
6	74.1848	83.8761
7	70.7207	90.0161
8	75.9921	89.9573
9	79.9569	92.7273
10	74.1870	91.8148

13- Experiments' Results of 10000 character-chunk

Case A

Run	Writer1	Writer2
1	92.0188	88.1127
2	92.5556	92.2860
3	94.8465	91.2192
4	96.2472	86.8568
5	91.5385	89.1732
6	90.9091	89.5349
7	92.9224	86.0731
8	92.8256	93.8063
9	87.9274	92.4292
10	91.7892	91.2990

Case B

Run	Writer1	Writer2
1	70.9231	72.0436
2	65.6452	85.1605
3	66.8939	72.6690
4	47.2109	66.3527
5	44.8387	67.2584
6	62.9310	77.6525
7	46.5806	68.5448
8	52.2727	69.3327
9	70.2703	66.8746
10	50.0633	73.5151

Case C

Run	Writer1	Writer2
1	73.2468	83.1917
2	72.1046	94.7774
3	57.2193	90.0615
4	81.5131	93.0601
5	75.0221	89.1628
6	68.0944	82.2034
7	79.3758	89.8438
8	83.5616	90.9628
9	80.0983	93.9542
10	78.7485	85.5705

Case D

Run	Writer1	Writer2
1	91.9000	94.2073
2	93.3654	96.8750
3	98.0847	98.2292
4	71.3542	61.9681
5	84.8140	96.9466
6	98.2824	88.3588
7	80.4762	82.1250
8	87.8472	97.4771
9	99.6377	96.4552
10	69.3182	91.7120

Case E

Run	Writer1	Writer2
1	67.5711	53.0222
2	59.1479	83.4586
3	59.1864	88.1778
4	75.5663	62.4018
5	59.1026	87.0811
6	83.0097	78.2728
7	68.7135	82.6520
8	74.3003	81.4374
9	92.4107	62.3344
10	59.7113	90.6362

14- Experiments' Results of Combining 1000 words chunk and 2000 words chunk

Case A

Run	Writer1	Writer2
1	95.2381	100.0000
2	73.1982	100.0000
3	53.2529	100.0000
4	94.9830	100.0000
5	83.3944	100.0000
6	81.9801	100.0000
7	93.9580	100.0000
8	96.4952	100.0000
9	64.3072	100.0000
10	79.1446	100.0000

Case B

Run	Writer1	Writer2
1	93.7211	95.4248
2	84.8073	96.7288
3	93.9578	98.9177
4	95.0951	98.3089
5	96.5797	88.4951
6	94.1787	59.9647
7	96.5769	60.3492
8	99.6887	85.5631
9	93.5185	95.1977
10	95.5867	66.8452

Case C

Run	Writer1	Writer2
1	89.7690	90.7407
2	95.1844	76.1336
3	94.0476	85.3276
4	87.0813	74.4244
5	94.2761	80.9960
6	91.4018	74.0569
7	96.5543	71.7794
8	95.5128	80.0281
9	87.1146	82.4228
10	94.1249	70.0547

Case D

Run	Writer1	Writer2
1	96.5706	74.4507
2	91.5278	83.9181
3	89.3577	53.4583
4	98.9040	90.0400
5	92.8753	96.9547
6	67.8687	86.6413
7	83.2099	91.3126
8	92.6275	97.0442
9	66.5705	88.6859
10	86.2434	93.6027

Case E

Run	Writer1	Writer2
1	94.6481	93.9606
2	76.5873	73.0769
3	84.8073	86.5202
4	89.1734	98.5559
5	91.4462	92.9664
6	81.6416	95.1671
7	95.4955	75.2245
8	89.7285	84.5967
9	87.6477	92.7063
10	91.8536	84.5679

15- Experiments results of combining Function words and character level features

Case A

Run	Writer1	Writer2
1	93.2018	100.0000
2	89.5556	100.0000
3	94.1725	100.0000
4	87.4728	100.0000
5	86.7089	100.0000
6	78.7500	100.0000
7	91.1890	100.0000
8	86.8653	100.0000
9	85.7298	100.0000
10	88.9597	100.0000

Case B

Run	Writer1	Writer2
1	85.5224	79.7721
2	84.5082	75.7251
3	72.9787	72.0477
4	64.1379	82.3955
5	80.8800	75.6824
6	90.1538	75.7085
7	85.9091	72.2507
8	81.1511	78.5755
9	70.9396	83.5759
10	82.1168	90.1338

Case C

Run	Writer1	Writer2
1	89.7450	79.5161
2	78.9112	94.6328
3	79.2727	79.5000
4	88.3117	87.5962
5	89.2889	78.2353
6	81.9825	78.9809
7	82.2100	78.7829
8	86.3046	84.1837
9	82.4355	80.0000
10	77.5218	88.2813

Case D

Run	Writer1	Writer2
1	90.3935	95.9091
2	89.2857	85.2564
3	97.3404	93.5688
4	87.0798	94.5697
5	83.5000	84.8039
6	97.4138	92.6095
7	97.3958	96.7949
8	97.5728	83.9286
9	96.8511	97.8545
10	90.8565	91.1215

Case E

Run	Writer1	Writer2
1	75.9690	90.3268
2	89.9123	96.9650
3	95.3125	77.4758
4	71.1667	83.2687
5	54.6205	97.3974
6	60.7937	91.1543
7	97.2574	49.5926
8	78.9683	83.1111
9	70.9571	89.9660
10	94.3333	67.2262

Appendix B

Publications

Investigating Hybrids of Evolutionary Search and Linear Discriminant Analysis for Authorship Attribution

Kareem Shaker, David Corne, and Richard Everson

***Abstract*— Authorship Attribution is the problem of determining who is (or was) the author of one or more texts, in cases where authorship is disputed. There are many well-known cases of disputed authorship; in this paper we consider the *Federalist Papers*, and the *15th Book of Oz*. We treat the problem as a supervised classification problem, and use evolutionary algorithms to search through subsets of *function words*, which in turn form the basis of predicting authorship via linear discriminant analysis. We compare two approaches (due to the size of the text corpora in dispute, extensive experimentation is difficult), both centred around the optimization of ROC curves. On both datasets, the hybrid EA approach was able to classify the disputed works with 100% accuracy, using small sets of function words comparable to or better than previous works on these cases.**

INTRODUCTION

THE Authorship Attribution problem is the challenging task of determining who was the author of a disputed text, given that two or more individuals claim sole authorship of that text. There are many historical examples relating to conflicting authorship claims. Two well-known cases, which we use as test cases in this paper, are the disputed Federalist papers [1] and the 15th Book of Oz [2].

As noted in a recent survey [3], a wide variety of approaches have been attempted for authorship attribution, but no specific approach emerges as well-regarded or well-used in this field. Typical approaches attempt to find patterns that characterise specific authors. These patterns may involve, for example, distributions of word lengths, of sentence lengths, and/or vocabulary distribution (in which is considered the overall diversity of terms used by an author).

A particularly popular and successful source of such characteristic patterns is the frequencies of so-called *function words*. This term was introduced in [1], in which Mosteller and Wallace suggested that an author's essential style could be characterised by the frequencies with which they used a relatively small number of specific words. In several studies since, function words have been found to be surprisingly useful in authorship attribution studies using a wide range of methods [4]. The primary list of 70 function words studied by Mosteller and Wallace [1] are provided in Table I.

Manuscript received April 6, 2007.

K. Shaker is with the School of Mathematics and Computer Science, Heriot-Watt University, Edinburgh, UK.

D. Corne is with the School of Mathematics and Computer Science, Heriot-Watt University, Edinburgh, UK, dwcorne@macs.hw.ac.uk

R. Everson is with the School of Engineering, Computer Science and Mathematics, University of Exeter, UK.

TABLE I: MOSTELLER AND WALLACE [1] PRIMARY FUNCTION WORDS.

a	all	also	an	and
any	are	as	at	be
been	but	by	can	do
down	even	every	for	from
had	has	have	her	his
if	in	into	is	it
its	may	more	must	my
no	not	now	of	on
one	only	or	our	shall
should	so	some	such	than
that	the	their	then	there
thing	this	to	up	upon
was	were	what	when	which
who	will	with	would	your

Typically, for a given authorship attribution task, researchers will choose a set of appropriate function words, and construct datasets consisting of frequency vectors for each of several texts known to be have been authored by the disputants. A statistical approach and/or a machine learning method is then used to train a classifier on these data, and this classifier is then applied to frequency vectors representing the disputed text(s). Many classification methods have been tried, ranging through neural networks [5] and support vector machines [6], joining a rich history of statistical and probabilistic approaches [e.g. 7, 8, 9].

A familiar theme in this research area is that of a semi-independent interest in both overall classification accuracy, and in the number of function words (and or other markers) required for an accurate classifier. Focus on the latter issue is of interest and importance in the wider science of stylometry [10], which considers the general question of numerical and statistical patterns that capture an author's (or a composer's) style. For example, in one study that compared several approaches on the Federalist Papers task [11], an evolutionary algorithm was employed to evolve rules that queried the frequencies of specific function words, and applied to the Federalist Papers task. Though not always successful in classifying the disputed papers, the results were promising and interesting in that rules were found that used a relatively small number of function words to achieve discrimination.

Given these paired themes of accuracy and feature selection, work on authorship attribution is beginning to emerge that combines the two in the expected variety of ways. For example, most work to date can be seen as involving *a priori* feature selection, in which researchers have pre-chosen a set of function words, and trained a classifier based on their frequencies. Meanwhile [12] uses an evolutionary algorithm for *combined* feature selection and classification, using the strategy (increasingly common in general) of a pre-chosen classifier (in their case, a support vector machine), and using an evolutionary algorithm to search the space of feature subsets for input to that classifier. The only other work we have found using evolutionary algorithms in this area is a preliminary report [13].

In this work we contribute a similar approach (that is new to the field of authorship attribution), in which we evolve feature subsets for classification by linear discriminant analysis (LDA), using area-under-ROC-curve as our fitness measure following the training of the classifier. The approach involves careful steps to ensure good generalisation performance, and we find that, for both the cases of the Federalist papers and the Book of Oz, this approach is able to yield discriminators that perfectly classify the disputed works, using numbers of function words than compare favourably with the literature.

The remainder is set out as follows. In section II we provide further background on Authorship Attribution, and in section III we expand on the *Federalist* and *Oz* datasets and the way that we process them. Section IV details the linear discriminant analysis (LDA) classifier that we use, with associated information about ROC curves, and section V describes two hybrid evolutionary algorithms that we use to evolve feature subsets for this classifier. Experiments and results are given in section VI, and we have a concluding discussion in section VII.

Further Background

In 1887, Mendenhall [14] reported one of the earliest known studies in the field of authorship attribution; he used word-length distributions to study certain works of John Stuart Mill, and compare these to work by others on the same topic. In 1901, Mendenhall then applied this method to works of Shakespeare and Bacon [15], however a recent examination of this work concludes [16] finds that the distinctions claimed by Mendenhall were mistaken, revealing distinctions in word-length distributions between poetry and prose, rather than between different author's styles.

Yule [17] provides the first examination of sentence-length for stylometry, characterising authorship in terms of, for example, mean and standard deviation of number of words per sentence. Seemingly successful in a variety of disputed-authorship cases, sentence-length based studies then became relatively frequent, e.g. [18]. However the more favoured approaches that emerged in the 60s and 70s were those based primarily on *function words* [1], and on vocabulary distribution. Vocabulary distribution [18, 19] measures the diversity of an author's vocabulary, and tends to involve mathematical models for the frequency distributions of the number of words appearing exactly *r* times (for example) for various *r*. Meanwhile, the use of function words was introduced in [1]; function words (table I) are specifically words that have no significant meaning, but play important grammatical and syntactic roles; they include pronouns, conjunctions, prepositions, auxiliary verbs, and some adverbs. Also, it is known that function words rarely borrow from other languages, and hence it is very rare for new ones to come into fashion, and hence rare for existing functions words to go out of fashion. These various characteristics of function words suggest that the way an individual authors uses them is dependent on his style, rather than affected by confounding factors such as age, era, content, and so on. It is interesting that this class of words are of great value in authorship attribution and similar studies, but are usually directly omitted (i.e. included in the list of *stop words* – e.g. see [20]) from text mining and related research concerning the *content* of documents.

As a result of the success achieved by Mosteller and Wallace [1], in using the relative frequencies of function words on the Federalist Papers task, research in their use has flourished, with a variety of classification methods having been studied, but each using function word frequencies as the stylometric 'fingerprint'. For example, function words were recently used to address the disputed Book of Oz [2], and the disputed Federalist Papers [21, 22].

Given the number of potential function words that can be employed (a 'primary' list of 70 is given in [1]), but also given the common scenario in which there are relatively few texts available for which to construct datasets (e.g. in the case of the Federalist papers, there are 66 data points), the authorship attribution problem involves serious challenges regarding overfitting. Consequently, dimension reduction strategies are common in the authorship attribution literature, with principal components analysis (PCA) often employed, along with exhaustive searches of small feature subsets [22]. Support vector machines are also beginning to be used in ths field, [6, 22], given their *a priori* expectation of good generalisation performance with relatively sparse data sets.

The Federalist Papers and Book of Oz data

The well-known Federalist Papers are a group of 85 essays, ranging between 900 and 3,500 words in length, all written under the same pseudonym, aimed at persuading the people of New York to approve the U.S. Constitution. 77 of the essays were published in newspapers between 1787 and 1788, and a further 8 were included when they were later published together as a book. The authors of the essays were Alexander Hamilton, James Madison and John Jay; it is known that Hamilton wrote 51 of them, Madison wrote 15 of them, and Jay wrote 5, with a further 3 being co-written by Hamilton and Madison, but a specific set of 11 papers are known as the *disputed* papers, for which both Madison and Hamilton claimed sole authorship. Hamilton's essays have a mean length of 2203 words, ranging between 987 and 5,733 words. Madison's mean number of words is 2755, ranging between 1,704 and 3575 words. Meanwhile the disputed papers average 2022, ranging between 1,133 and 3056. The frequency of the 70 function words is calculated for each individual paper, producing 77 vectors (51 Hamilton, 15 Madison, and 11 disputed). The data are downloadable from [23] and [24].

Two authors, Lyman Frank Baum and Ruth Plumly Thompson wrote the 33 *Adventure Oz* tales. Baum began writing in 1900 and is known to have written at least 14 tales before his death. Then Thompson took over and continued the tales until tale number 33. The *15th Book of Oz* was published under Baum's name, one year after his death, and Thompson then claimed that she was the sole author of the 15th Book of Oz.

The 14 tales by Baum and 5 of Thompson's tales can be easily found in electronic form, from [25] and [26] respectively. The average number of words in Thompson's tales is 39,017 words, ranging between 33,842 and 45,654 words. The corresponding figures for Baum are 42,100, 38,413 and 53,206. We partitioned each book into several data points by considering pairs of chapters as a unit. In the end, this provided 86 function-word frequency vectors for Baum, 49 for Thompson, and 12 for the single disputed book.

Finally, we note that in both cases, as a result of the wealth of evidence from authorship attribution studies, combined with further historical research, neither the disputed Federalist papers nor the disputed book of Oz is actually *disputed* anymore. It is generally accepted that the disputed papers are the work of Madison, and the disputed book is the work of Thompson. From the viewpoint of further authorship attribution research studies, this makes both datasets rather less exciting. However, of course it also makes it possible to evaluate and test authorship attribution methods, since we are able to evaluate the accuracy of the results, and consequently these works continue to be used in this research area. We mention finally that the thrust of our own research is, having developed techniques on this and similar test cases, to apply the attribution methods to well-known and historic still-disputed works in the first-authors' culture.

The Linear Discriminant Analysis Classifier

Given relatively high dimensionality of data compared to the number of data points available in each case, and also given the unbalanced class sizes (51 vs 14 in the Federalist papers case, and 49 vs 12 in the Book of Oz case), we chose to use Linear Discriminant Analysis (LDA) for the classifier (an accessible tutorial is at [27]). LDA naturally and appropriately handles the problems of unequal class sizes. It works simply by finding a linear function of the data vectors that defines a separating hyperplane which separates the data as well as possible, specifically aiming to minimise the ratio of within-class variance to between-class variance. The weights for the discriminating hyperplane are learned by minimizing the cross-entropy error function. Meanwhile, to promote good generalization performance, we use leave-one-out cross-validation, and weight-decay regularization. Weight decay regularization attempts to keep low the absolute values of the weights in the discriminant function, which in turn tends to be associated with better generalization performance, however it involves a parameter which is difficult to choose correctly in advance. We thus repeat the LDA training process several times for different values of this parameter.

Our LDA training implementation uses the Netlab library [28]. The step by step procedure is as follows. We are given a set of vectors to classify (in this paper, all problems are two-class classification problems). Suppose there are m d-dimensional vectors (of function-word frequencies) in the training set; we will denote the ith such vector as $\boldsymbol{v_i}$, with elements $v_{i,1}$, $v_{i,2}$, …, $v_{i,d}$. An LDA classifier is trained, learning a vector of weights $\boldsymbol{w} = (w_1, w_2, \ldots, w_d)$ which minimises an error function $E = E_c + E_d$. This error function is a combination of the cross-entropy error term and the weight-decay regularisation term, where respectively:

$$E_c = -\sum_{i=1}^{m} t_i \ln(v_i \cdot w) + (1 - t_i) \ln(1 - v_i \cdot w)$$

and

$$E_d = \alpha(w \cdot w)$$

where t_i is either 0 or 1, denoting the correct class value of vector v_i.

Training is done via the iterative re-weighted least squares algorithm, using default parameters in the Netlab implementation. This training process is repeated for a range of different values of α, estimating the quality for each value by the average performance of the trained LDA using leave-one-out-cross-validation.

Thus, the input to the LDA process is a set of classified training examples, and the output is a classifier, the one corresponding to the (or a) best α value. The resulting classifier is then used in the following way.

Given a *d*-dimensional test vector, *x*, its classification is indicated by the dot product $x \cdot w$. We convert this into a value between 0 and 1 using the logistic equation, and hence record the value:

$$c_i = 1 \Big/ 1 + e^{-x_i \cdot w}$$

for each individual vector x_i of a set of test vectors. This is not vital, but convenient for the next step, which is to compute the ROC curve for the classifier, by calculating the false positives and true positives ratios on the test set for each of several threshold values between 0 and 1. In other words, the output of the classifier is a number between 0 and 1, where the expectation is (for example, where just two authors are involved) that texts written by one author will lead to an output closer to 0, and texts written by another will lead to an output closer to 1. Any particular threshold *t* between 0 and 1 will lead to a point on the ROC curve. For example, setting the threshold at 0.3 indicates that we class test inputs as being written by author *A* if the output is below 0.3, and by author *B* if the output is above 0.3. This leads to a specific pair of points on the curve (proportion of correctly classified author *A* texts plotted against proportion of correctly classified author *B* texts). A collection of thresholds therefore leads to a curve. One measure of the classifier's performance is the area under this curve. An area of 1 indicates perfect performance.

The Hybrid ROC-dominance based Approach

In addition to straightforward principal components analysis (PCA), we report here the investigation of two algorithms that hybridise simple evolutionary algorithms with the LDA training process described above. In this section we describe the first of these, which wraps a simple evolutionary algorithm around the LDA process, but using an idea from [29] as a non-standard way to select parents, based on using an archive of non-dominated ROC curves.

The basic procedure is as follows. Each chromosome encodes a non-empty (but otherwise unrestricted) subset of the 70 primary function words from [1]. The encoding used is simply a list of features. The fitness of a chromosome in this case is its ROC curve – hence this a multiobjective approach [30—32] (although we do not yet employ any sophisticated or up-to-date strategies from the multiobjective evolutionary algorithm literature in this work). This is obtained by running the LDA process described above on the training data to produce the best classifier, and generating the ROC curve from that classifier.

Initially, we start with a population of one, which represents a subset of size one – i.e. the initial chromosome encodes a singleton, representing a randomly chosen one of the 70 function words. More generally, while the algorithm is running, there is an archive of chromosomes maintained, which is mutually nondominated w.r.t. their ROC curves. No size limit is enforced for this archive (see discussion about such issues in [33]).

The algorithm proceeds as follows, following the generation, evaluation and archiving of the initial random solution. For *gen* iterations, we select a chromosome from the archive, and then randomly choose to either *add*, *delete*, or *change* a randomly chosen feature (naturally, only valid choices are made, so that we do not delete a feature from a singleton set, or duplicate an existing feature, etc.). The resultant mutant is evaluated, and then the archive is appropriately updated.

The Hybrid AUC-Fitness Approach

Our second approach also wraps a simple EA around the LDA training process, but this times simply calculates the area under the ROC curve (AUC), and treats this as a single-objective fitness value to be maximised. For this approach we use a straightforward small-population steady-state evolutionary algorithm. Specifically, population size 5, mutation (only) operator as described in section V, binary tournament selection, and replace-worst replacement, breaking ties by number of features. That is, in each generation, binary tournament selection is used to choose a parent. A mutant is then generated and evaluated. The mutant enters the population if it is at least as fit as the current worst. If there is a tie between the mutant fitness and the fitness of the current worst, but the mutant contains more features than the current worst, then the mutant is discarded.

As with the ROC-dominance approach, the initial population contains only randomly chosen singleton feature vectors.

Experiments and Results

For each of the Federalist Papers and Book of Oz in turn, we will now present the results of preliminary analysis with PCA, and then the two hybrid approaches.

Federalist Papers: Principal Components Analysis

Following standard PCA applied to the Federalist Papers data, we plotted the projections of the data onto the first two principal components in Figure 1. It can be seen that the general positions of the eleven disputed papers (squares) are not able to be clearly distinguished from either the Madison papers (crosses) or the Hamilton papers (squares). Arguably, they are shifted more towards the 'Madison space' than the 'Hamilton space', but several individuals are much closer to Hamilton papers than to any Madison paper.

Following PCA, we used the principal component transformations of the data as input to the LDA process already described, trying this for the first *k* principal compoents, for each *k* from 2 to 70. For each such *k*, we measured validation error by recording the cross-entropy error on the disputed papers. Figure 2 shows the plot of validation error against number of principal components used. Clearly, the findings from PCA are that we need the first 15 principal components to reach the minimum error, tentatively suggesting that around that many function words may be needed (in the sense that this is the suggested number of latent features required for good performance).

We ran each of the Hybrid ROC-dominance approach of section V (ROCD) and the Hybrid AUC-fitness approach (AUCF) 10 times with the following parameters. We performed 300-iteration runs, where each evaluation incorporated 5 runs of the LDA training process for different values of α randomly chosen between 0.1 and 1. AUCF used a population size of 5. Results are summarized in table II. The table reports results on the *training set*, but not on the disputed papers. On the disputed papers, the best of 10 trials for 300 iterations was a set of 3 function words.

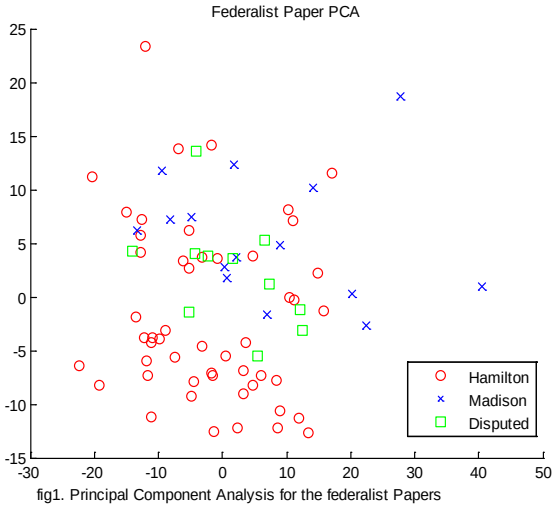


Figure 1: Projection of the first two principal components of the Federalist Papers data.

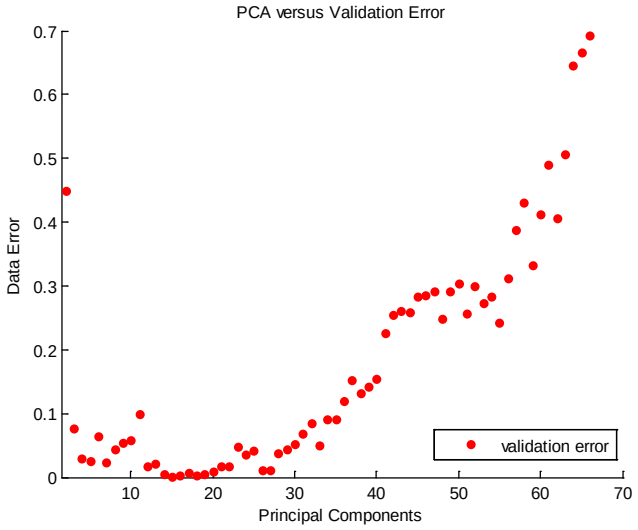


Figure 2: Error on the disputed papers against number of principal components learned from the test data: Federalist papers.

Federalist Papers: The Hybrid ROC-Dominance Method and the Hybrid AUC-Fitness Approach

There is insufficient evidence so far to support a statistical claim that AUCF is a better approach than HROC on this problem, however the important and interesting findings are that these methods can both reliably obtain classifiers that use only 2 function words. In the case of AUCF, the LDA classifiers associated with each best-fitness set of words emerging from the training process was tested six times on the training data with randomly perturbed values of the regularization parameter. When the training set classification was perfect each time, this set of function words (and its associated classifier) was regarded as *stable*, and used on the unseen test set of disputed papers. In all cases, such a stable set of function words also achieved perfect discrimination on the disputed papers. None of the 2-function word sets was stable, however the best of 10 trials at 300 cycles found a stable set of 3 words. Unfortunately, at the time of writing we have not been able to collect the corresponding test results for HROC.

TABLE II: HROC/AUCF METHODS TRAINING RESULTS ON FED. PAPERS.

	HROC	AUCF
Best of 10 trials after 100 cycles	4 function words, achieving perfect discrimination.	**2** function words, achieving perfect discrimination
Mean of 10 trials after 100 cycles	4 function words, achieving perfect discrimination.	**2** function words, achieving perfect discrimination
Best of 10 trials after 300 cycles	**2** function words, achieving perfect discrimination	**2** function words, achieving perfect discrimination
Mean of 10 trials after 300 cycles	3.1 funct. words achieving perfect discrimination	**2** function words, achieving perfect discrimination

In comparison, Fung [6], using support vector machines, found a classifier that also used 3 function words (*to*, *upon*, and *would*), while Bosch and Smith [22] achieved the same result with an extensive test that searched all combinations of 1, 2 and 3 function words using a linear programming

formulation, discovering a single set of 3 (*as*, *our* and *upon*) that achieved perfect classification. In our case, HROC was able to find a classifier that worked only with *such*, *upon* and *with*, while different perfectly-classifying sets of words were found by AUCF, but usually including the word *upon*.

The Book of Oz: Principal Components Analysis

Figures 3 and 4 respectively show the projections of the first two principal components for the *Book of* Oz data, as follows. In figure 3, each tale is a separate point, while in figure 4 we plot the 86 points obtained by dividing the tales into two-chapter chunks. 'Thompson space' is represented by the crosses, Baum is represented by circles, and the disputed book is represented by squares.

Again, it is not clear from PCA alone to whom authorship of the disputed book should be attributed. These plots do seem to lean a little towards Thompson, however far from enough to have any real confidence in that conclusion. When using the principal components vectors in tandem with the LDA classifier, we found that the best validation error was obtained when using the first 26 principal components (compare with 15 for the Federalist papers). This provides evidence that the attribution task for the Book of Oz is more complicated than for the Federalist Papers, needing correspondingly more features to obtain a good distinction between the two authors' writing styles. The corresponding plot of validation error against number of components is in Figure 5.

Book of Oz: HROC and AUCF

We ran each of the Hybrid ROC-dominance approach of section V (ROCD) and the Hybrid AUC-fitness approach (AUCF) 10 times with the same parameters as described in section VII.B. Training results are summarized in Table III; meanwhile, using the same approach to choosing sets of words for analysis of the (unseen) disputed works, we again found that *stable* best-performing classifiers from the AUCF training runs always produced perfect results on the test set. The best of the 10 runs at 300 iterations found a stable set of 6 words.

Again, there is insufficient evidence so far to support a statistical claim that AUCF is a better approach than HROC on this problem, however we again have interesting findings that show that each method is adept at reliably discovering relatively small subsets of features that can perform perfected discrimination of the disputed work. Previous work on this case is less common than in the case of the Federalist papers, and we do not have comparable results concerning attempts to minimise the number of features. Binongo [2], concentrated on using principal components of a set of 50 function words. We feel it is therefore an interesting contribution that we have found sets of six words that can lead to perfect discrimination on the (unseen) disputed works.

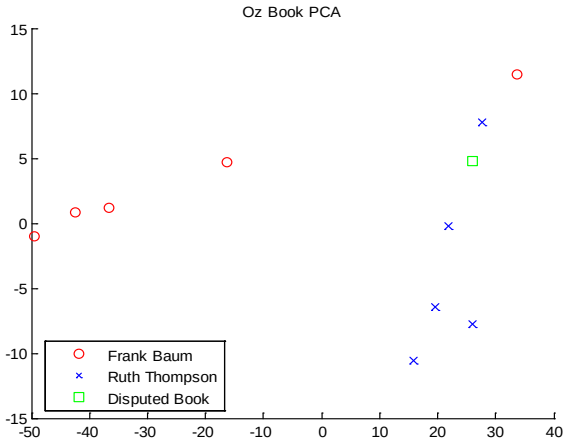


Figure 3: Projection of the first two principal components of the Book of Oz data – one point per tale.

Figure 4: Projection of the first two principal components of the Book of Oz data – one point per pair of chapters.

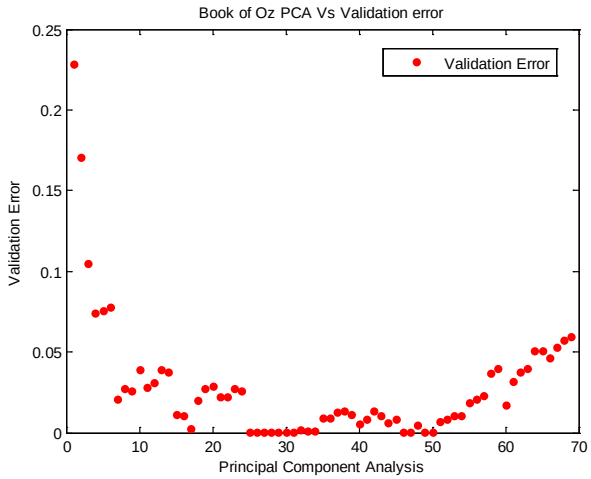


Figure 5: Error on the disputed papers against number of principal components learned from the test data: Book of Oz.

TABLE III: HROC/ AUCF METHODS TRAINING RESULTS ON BOOK OF OZ.

	HROC	AUCF
Best of 10 trials after 100 cycles	6 function words, achieving perfect discrimination.	**5** function words, achieving perfect discrimination
Mean of 10 trials after 100 cycles	6.5 function words, achieving perfect discrimination.	5.8 function words, achieving perfect discrimination
Best of 10 trials after 300 cycles	6 function words, achieving perfect discrimination	**5** function words, achieving perfect discrimination
Mean of 10 trials after 300 cycles	Perfect discrimination always achieved, with mean of 6 function words	**5** function words, achieving perfect discrimination

Concluding Discussion

In conclusion, we addressed the problem of Authorship Attribution using hybrids of simple evolutionary search and a linear discriminant classifier, using evolutionary search to find small function word subsets as the sets of features used in training the classifier. Baseline comparison was also done with using straightforward PCA to transform the data, finding that 15 and 26 components were needed respectively to obtain perfect performance on the disputed *Federalist Papers* and *Book of* Oz respectively. Using a simple EA to search feature subsets based on iteratively selecting randomly from subsets with so-far nondominated ROC curves (the HROC approach), we were able to reliably find subsets of function words of sizes 3 and 6 respectively. Regarding the Federalist Papers task, this equals what has been achieved before in the literature, which in turn has some implications and interest for stylometry studies. In the case of the much more difficult *Book of Oz* case, we can only conclude that the result seems very good given the large number of principal components required, and perhaps sets a target for related studies, since work so far has not used the *Book of Oz* task in an explicit attempt to minimise the number of function words used for discrimination. A simple EA for evolving ROC curves, again hybridised with the LDA classifier (which we called AUCF), achived slightly better results here during training than the HROC approach.

Our work on this so far has been hampered by the long training times required by the LDA classifier built into our fitness function, and the corresponding repeated runs of that process that are required to find a good parameter for the weight decay regularisation. In ongoing work we will compare this with less time-consuming classifiers, and so it has yet to be seen whether similar or better results can be achieved with a less sophisticated classifier. Finally, it is clear that evolutionary algorithms have a potential role in authorship attribution, and stylometry in general, particularly regarding feature selection.

REFERENCES

[1] Mosteller, F. and Wallace, D.L., *Inference and Disputed Authorship: The Federalist*, Reading:Addison-Wesley, 1964.

[2] José Binongo., Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution, *Chance*, **16**(2):9—17, 2003.

[3] Patrick Juola, John Sofko, and Patrick Brennan. (2006). A Prototype for Authorship Attribution Studies. *Literary and Linguistic Computing* **21**:169-178

[4] Shlomo Argamon, Shlomo Levitan (2005). Measuring the Usefulness of Function Words for Authorship Attribution. Association for Literary and Linguistic Computing/ Association Computer Humanities, University Of Victoria, Canada.

[5] Kjell, B. (1994). Authorship Determination Using Letter-pair Frequency Features with Neural Network Classifiers. *Literary and Linguistic Computing, 9*, 119 – 124

[6] G. Fung and M. Olvi (2003) The Disputed Federalist Papers: SVM Feature Selection via Concave Minimization (2003).

[7] Brainerd, B. (1975). Statistical anlaysis of Lexical data using Chi-squared and related distributions. *Computers and the Humanities, 9*, 161 – 178

[8] Burrows, J. F. (1987). Word Patterns and Story Shapes: The Statistical Analysis of Narrative Style. *Literary and Linguistic Computing, 2*, 61 -67.

[9] Doležel, L., & Bailey, R. W. (Eds.). (1969). *Statistics and Style*. New York: Elsevier.

[10] D. I. Holmes (1998) The Evolution of Stylometry in Humanities Scholarship, Literary and Linguistic Computing, **13**(3):111-117

[11] D. I. Holmes and R. S. Forsyth (1995) The *Federalist* Revisited: New Directions in Authorship Attribution, *Literary and Linguistic Computing*, **10**(2):111-127

[12] Jiexun Li, Rong Zheng and Hsinchun Chen (2006) From fingerprint to writeprint, *Communications of the ACM*, **49**(4):76—82.

[13] S.G. Efimovich and S.O. Gennadyevich (2003) Automatic search of indicators of text authorship, *Proceedings of the Korea-Russia International Symposium – KORUS 2003*: 185—188.

[14] Mendenhall, T.C. (1887) The Characteristic Curves of Composition, *Science*, IX, 237-249.

[15] Mendenhall, T.C. (1901) A mechanical solution to a literary problem, *Pop. Sci. Monthly*, **60**: 97—105.

[16] Williams, C.B. (1975) Mendenhall's studies of word-length distribution in the works of Shakespeare and Bacon, *Biometrika*, **62**(1): 207—212.

[17] Yule, G.U. (1939) On sentence-length as a statistical characteristic of style in prose, with applications to two cases of disputed authorship, *Biometrika* **30**:363—390.

[18] Morton, A.Q., Literary Detection, New York:Scribners, (1978).

[19] Sichel, H.S., "On a Distribution Law for Word Frequencies", Journal of the American Statistical Association, 70, 542-547, (1975).

[20] C. J. vanRijsbergen. Information Retneval. 2nd ed., Butter-. worths, 1979.

[21] Bradley, P. S. and Mangasarian, O. L., Feature selection via concave minimization and support vector machines. In Machine Learning Proceedings of the Fifteenth International Conference(ICML '98), J. Shavlik, Ed. Morgan Kaufmann, San Francisco, California, 82-90.(1998)

[22] Bosch, R. A. and Smith, J. A. Separating hyperplanes and the authorship of the disputed Federalist papers. American Mathematical Monthly 105, 7 , 601-608 (1998).

[23] http://www.cnsnews.com/Library/Federalist/Default.htm
[24] http://thomas.loc.gov/home/histdox/fedpapers.html
[25] http://www.literature.org/authors/baum-l-frank
[26] http://onlineBooks.library.upenn.edu
[27] http://marketing.byu.edu/htmlpages/tutorials/discriminant.htm
[28] http://www.fizyka.umk.pl/netlab/index.htm
[29] Richard M. E and Jonathan E. F, Multi-Objective Optimisation of Safety Related Systems: An Application to Short Term Conflict Alert (2004)
[30] C. Coello, "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques," *Knowledge and Information Systems. An International Journal*, vol. 1, no. 3, pp. 269–308, (1999.)
[31] Veldhuizen D. V. and Lamont G., "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation* vol. 8, no. 2, pp. 125–147, (2000.)
[32] Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: Wiley, (2001)
[33] Fieldsend, J., Everson, R., and Singh, S. Using unconstrained elite archives for multiobjective optimization, *IEEE Trans on EC*, **7**(3):305—323.

Authorship Attribution in Arabic using a Hybrid of Evolutionary Search and Linear Discriminant Analysis

Kareem Shaker and *David Corne*
School of Mathematical and Computer Sciences, Heriot-Watt University
Edinburgh, EH14 4AS, UK
ks113@hw.ac.uk, *d.w.corne@hw.ac.uk*

Abstract

Authorship Attribution is the problem of determining the authorship of one or more texts. Applications include disputed authorship, or deciding which of a collection of pieces of text were by the same author. A popular and successful approach is to characterize a specific author in terms of the usage pattern of function words. These are common words that are unrelated to subject matter, and tend to be used in specific ways by different authors. In English, a well-known collection of 70 function words is often used for this purpose. Previously, using a hybrid of evolutionary search and linear-discriminant analysis (LDA), we have shown excellent performance in authorship attribution in English based on a function word approach. Here, for the first time, we propose and test a set of Arabic function words for use in Arabic authorship attribution. Tests indicate that the chosen collection forms an effective basis for authorship attribution in Arabic.

1. Introduction

The Authorship Attribution problem is the task of determining the authorship of a given piece of text. In cases of disputed authorship, two (or maybe more) distinct

individuals may claim authorship, and there are several historical examples of such conflicting authorship claims. For example, two well-known cases of disputed texts in English include the disputed Federalist papers [1] and the 15th Book of Oz [2].

A wide variety of methods have been researched for authorship attribution (e.g. see [3] for a survey). The main issue of interest is how to represent an author's 'fingerprint', which overlaps almost completely with the issue of how to encode a piece of text as a feature vector. A subsidiary issue is the choice of machine learning method that will then be used to produce classifiers, that will in turn attempt to predict authorship for disputed stretches of text. As yet there is no clear convergence on any particular encoding or machine learning approaches, but a certain approach to the encoding of text is particularly popular and successful; this relates to the use of *function words*.

Table I: Mosteller and Wallace [1] function words.

a	all	also	an	and
any	are	as	at	be
been	but	by	can	do
down	even	every	for	from
had	has	have	her	his
if	in	into	is	it
its	may	more	must	my
no	not	now	of	on
one	only	or	our	shall
should	so	some	such	than
that	the	their	then	there
thing	this	to	up	upon
was	were	what	when	which
who	will	with	would	your

The use of so-called *function words* for authorship attribution was introduced by Mosteller and Wallace [1]. The idea is that an author's style can be characterised in terms of the frequencies with which that auhor uses each of a relatively small number of specific words. These are 'function words' in the sense that their use should be independent of the content or subject matter of any given text. E.g. if a writer writes one essay about cars, and another essay about flowers, these two essays will show quite different overall distributions of words, however when only the distributions of the *function words* are considered, we might expect no significant differences for these two essays, given that they were written by the same author. However, the hypothesis behind function words is that there may be significant differences in the function word usage between *different* authors. This hypothesis has been borne out in several studies [4]. The primary list of 70 function words used by Mosteller and Wallace [1] is provided in Table I.

Typically, given an authorship attribution or related task, researchers will choose a set of function words, and construct datasets consisting of frequency vectors of function word usage, for each of several sections of texts with known authorship. A statistical and/or a machine learning method is then applied to these data, yielding a classifier. The classifier is then applied to test data, which, in a real case of disputed authorship, will be function word frequency vectors associated with the disputed text(s). The familiar range of classification methods have been attempted, including neural networks [5], support vector machines [6], and various statistical and probabilistic approaches [7, 8, 9], including linear discriminant analysis [17, 19] and evolutionary search [11, 12, 13, 17].

In recent work [17] we explored a hybrid of evolutionary search with linear discriminant analysis for authorship attribution in English, where the emphasis was on attempting to find minimal stylistic fingerprints (i.e. a small set of function words) that were sufficient for the cases studied. In that work, we evolved feature subsets for classification by linear discriminant analysis (LDA), using the area under the ROC-curve (Receiver Operator Characteristic) as our fitness measure following the training of the classifier. This hybrid EA/LDA approach, which we use here, involves steps to ensure good generalisation performance in the parameterisation of the LDA, and found excellent results for the celebrated disputed authorship (in English) cases that were studied (the Federalist papers and the Book of Oz). The approach was particularly good at finding minimal subsets of English function words that could support accurate classification; this is of particular interest in the general study of stylometry [10]. However in this paper we are only concerned with predictive accuracy.

Finally, we note that there has been very little study of function words in alternative languages, and certainly no studies can be found that attempt to posit and test function words for authorship attribution in Arabic. In this paper we introduce and test function words in Arabic, motivated in part by a number of disputed authorship scenarios in the Arabic religious literature (although it is proving hard work to obtain the associated texts in electronic form). There are clearly other applications for an Arabic function word set, including (as with any language) questions of stylistic analysis, plagiarism investigations, and other investigations.

The remainder is set out as follows. In section 2 we provide further background on Authorship Attribution and function words, while in section 3 we expand on our hybrid EA/LDA classifier. Section 4 describes our dataset of Arabic novels, and section

5 reports experiments and results that refined and tested sets of Arabic function words. We summarise and discuss in section 6.

2. Background and Related Work

Authorship attribution studies began in 1887, when Mendenhall [14] reported using word-length distributions to study certain of the works of John Stuart Mill, comparing them to work by others on the same topic. Mendenhall followed this up in 1901, by applying his method to certain works of Shakespeare and of Bacon [15]. Though seminal, Mendenhall's work can be criticised [16] for mistakenly revealing differences in word-length distributions between poetry and prose, rather than between different author's styles.

The first to examine sentence-length distribution (rather than word-length) was Yule [17], who attempted to characterise authorship in terms of, for example, mean and standard deviation of number of words per sentence. This method showed some success, leading to more sentence-length based studies [18], however such studies were supplanted in the 1960s by characterisations in terms of function words [1], and more generally on vocabulary distribution [18, 19], which measures the diversity of an author's vocabulary. Typically, vocabulary distribution is modelled by frequency distributions of the number of words appearing exactly *r* times (for example) for various *r*. The function words approach [1], in contrast, deals only with specific words (such as pronouns, conjunctions, prepositions, and so forth) that have no significant meaning, but are grammatically and syntactically important.

Research in function word based authorship attribution flourished (e.g. [2, 21, 22]) following Mosteller and Wallace's demonstration of their use for the case of the disputed Federalist Papers [1]. Work in this area still tends overwhelmingly to concern English texts, and focuses on the comparison of different learning methods and/or augmentations to a function word approach [e.g. 6, 22].

3. The EA/LDA Classifier

One of the main challenges for machine learning approaches in authorship attribution is the typically small size of datasets in terms of number of samples (frequency vectors). Methods that incorporate careful handling of over-fitting are therefore common, such as linear discriminant analysis [27] and support vector machines. In this article we use the hybrid Evolutionary Algorithm / Linear Discriminant Analysis classifer (EA/LDA) described in [17], using the variant that evaluates fitness using the area under the ROC curve returned by the LDA classifier. The role of the EA is to find a subset of the function words that are in turn used to train the LDA. The LDA works simply by finding a linear function of the frequency vectors that defines a hyperplane which separates the data as well as possible, minimising the ratio of within-class variance to between-class variance. The weights for the discriminating hyperplane are learned by minimizing the cross-entropy error function.

In straightforward terms, the role of the LDA classifier is as follows. First, a subset of function words is supplied by the EA (i.e. a chromosome defines a subset of the function words). The input to the LDA is then the set of labelled training vectors, reduced in dimension (i.e. retaining only the elements indicated in the EA chromosome). The LDA then learns a weight vector, characterising a good separating hyperplane for the two classes (authors). The weight vector is then used for classification simply via considering its dot product with a test vector. The dot products are transformed between 0 and 1 by the logistic equation, and this value essentially represents a fuzzy decision (with one author associated with 0, and the other associated with 1). Following consideration of all test inputs (leave-one-out cross-validation is used), by considering a series of threshold values, an ROC curve is then constructed (equivalently, the curve indicating the tradeoff profile between false positives and true negatives).

Further detail is given in [17], while an accessible reference to LDA is in [27], and our LDA uses the Netlab library [28]. Finally, we describe aspects of the Evolutionary Algorithm (EA) used. A simple EA is wrapped around the LDA training process; as indicated, the EA supplies the chromosome (a subset of the available function words) and the LDA evaluates it, supplying in the end an ROC curve summarizing performance on the accumulated validation cases during leave-one-out cross-validation. The EA simply uses the area under this curve (AUC) as the fitness value to be maximised. As in [17], the EA is otherwise a straightforward steady-state evolutionary algorithm with a population size of 5, a mutation (only) operator that, with equal probability, either deletes a random feature, changes a feature, or includes a new feature, binary tournament selection, and replace-worst replacement, breaking ties by number of features (preferring fewer). In each generation, a parent is chosen via binary tournament selection, a mutant is then generated and evaluated, and then enters the population if it is no worse than the current worst. In the case of a tie between mutant and current worst fitness, the mutant is retained only if it does not contain more features than the current worst. Finally, the initial population contains only randomly chosen singleton feature vectors.

4. Arabic Text Datasets

To derive and test a collection of Arabic function words we procured a dataset of 14 books by six different writers. These were obtained from the website of the Arab Writers Union (www.amu-dam.net). The books ranged in size from 13,987 words to 37,567 words, with a mean of 23,942 words. Each of these books (details in Table II) were downloaded and processed to convert them into a string of Arabic words without extraneous characters and spaces.

5. Experiments: Refining the Arabic Function-Word Set

The initial set of Arabic function words was based on creating a collection of common prepositions and conjunctions, mirroring the semantic structure of the Mosteller and Wallace set for English. This led to a collection of 106 Arabic words.

We then investigated the frequencies of each of these 104 words over the complete collection of 14 books. This revealed that around 40 of the words were particularly common among all of the books, with patterns of usage that (on first sight) appeared roughly uniform, while a further 40 of these words tended to be of particularly low frequency. In preliminary work not reported here in detail, we used the test cases (Table III) to evaluate two subsets of words in turn. First, a collection of 64 words (omitting only the most frequent ones), and secondly a collection of 65 words (omitting only the least frequent). Only the latter set was found to be particularly promising, and we include results from this collection of 65 words below. Meanwhile, this set (which we call AFW65) is given in Table IV, which gives a numeric ID, the Arabic representation, and (in most cases) a ‘ballpark’ translation into English.

Table II: Details of the Arabic books dataset; includes shortened IDs for books and authors used later in presentations of results.

Authors	**Book details and IDs**
Ibrahim Khalil	Haris Al Maiz (HAM: 14,679 words
	Sodom Sebake Al Awez (SSAA: 2 parts, 28156 words and 29518 words)
Basem Ibrahim Abdo	Gesr Al Mawt (GAM, 37567 words
	Zahra fi Al Remal (ZFAR: 2 parts, 23241 words and 26061 words)
Taleb Omran	Ahzan Al Sinbad (AAS, 20389 words)
	AlBood Al Khamis (AAK, 13987 words)
	Madina Kharig Al Zaman (MKAZ, 14513 words)
	Al Fetiah Al Aghrar we Asfar al Kashf (AFAA, 19063 words)
Mary Show	Defly (DE, 24062 words)
	Awel Hob and Akheir Hob (AHAH: 2 parts, 18848 words and 19807 words)
Mohamed Youssef Salibi	Al Taih (AT, 36892 words)
	Sebahaa fi Al Wahl (SFAW: 2 parts, 25647 words and 28274 words)
Hessen Abd Al Kareem	Al Nabaa (AN: 2 Parts 29644 and 29472 words)
	Shagaret al Toot (SAT: 2 parts, 19024 words and 19998 words)

Table III: Details of the five test cases used in experiments; in each case, books from two authors constitute the training set, and different books from the same two authors comprise the test set.

Test case	**Train and test set details**
A	Training set: Books AAK and HAM Test set: Books MKAZ and SSAA
B	Training set: Books AHAH and SATP1 Test set: Books DE and ANP1
C	Training set: Books GAM and SFAWP2 Test set: Books ZFARP1 and AT
D	Training set: Books ANP1 and AAS Test set: Books ANP1 and AFAA
E	Training set: Books AFAA and ZFARP2 Test set: Books MKAZ and GAM

Table IV: AFW65 – a set of 65 Arabic function words, constructed by positing a collection of 104 candidate words, and removing those with low frequency in a collection of Arabic books

1 في	2 من	3 عن	4 على	5 إلى
in	From	about	over	to
6 حتى	7 فلا	8 منذ	9 لا	10 ثم
till	Not	since	no	then
11 بل	12 لكن	13 أو	14 أم	15 أن
	But	Or	Or	that
16 كأن	17 إن	18 إذن	*19* كي	20 لن
as if	The	therefore	So that	not

21 لم	22 ما	23 أي	24 ألا	25 أما
no	What	Any	Not	
26 ها	27 إذ	28 إذا	29 لو	30 لولا
		If	If	
31 هل	32 يا	33 نعم	34 بلا	35 هذا
Is?	Oh	Yes	without	this
36 ذلك	37 هذه	38 تلك	39 هؤلاء	40 أولئك
that	this	Such	they	those
41 الذي	42 التي	43 الذين	44 هو	45 هم
whom	which	Whose	he	them
46 هي	47 أنت	48 أنتم	49 أنا	50 نحن
she	You	you are	I	we
51 الآن	52 بين	53 هنا	54 هناك	55 كان
now	between	Here	there	been
56 ليس	57 أصبح	58 ظل	59 ماذا	60 لماذا
not	became	Keep	what	why
61 كيف	62 كم	63 أين	64 متى	65 مهما
how	how many	Where	when	whatever

At this point, some preliminary notes about experimental setup is in order. The common way in which a set of function words is employed (and which we do here) is to transform a section of text (a 'chunk') into a vector of *n* numbers in the interval [0, 1], each indicating the frequency of a function word as a proportion of the total words in that chunk. That is, if the chunk of text contains 1,000 words, and element *i* of this vector is 0.022, this indicates that function word *i* occurs 22 times in that chunk. To formulate an authorship attribution problem (or simply a simulated such problem) as a data mining task, a large section of text, such as a book, is partitioned into chunks of *c* words, for some *c*, and a frequency vector is built for each chunk. Each such frequency vector then is then associated with a target class, which in turn is simply the author of that chunk.

Hence, in each test case, an authorship dispute is simulated by supposing that we have two authors, Author1 and Author2, who both claim to have written each of Book1 and Book2. The training set comprises an undisputed book from each of the two authors, while Book1 and Book2 comprise the test set. In test case A, for example (see Table III), the training set comprises XXX chunks from the book ABC and YYY chunks from the book BCD, and the test set comprises XX chunks from the book CDE and YY chunks from the book DEF.

An important consideration is the size of the chunks. A feel for this can be gained from considering the extremes. If the chunk sizes were very small, function word frequencies would generally be very low, and often zero, and we would expect that each chunk would be too small to capture the stylistic fingerprint of an author. If the chunk sizes were very large (e.g. we could transform an entire book into a single frequency vector), we would have too few samples to do reliable machine learning, and could expect poor generalization performance.

A number of preliminary experiments were done with different chunk sizes, and we report here the more successful sizes, which were 1,000 and 2,000 words respectively. Table V summarises results on the five test cases for the function word set AFW65, for each of 1,000 and 2,000 word chunks. Each entry in the table corresponds to the mean of five trial runs, each of which ran for a specific parameterization of the hybrid classifier (determined in advance from preliminary experiments, and similar to the configuration that achieved best results for English authorship attribution in [17]). The result of an experiment is a percentage accuracy figure, which indicates the

percentage of chunks in the test set that were correctly labeled by the classifier; this figure is always the mean over 5 trials. Notice that, in one sense, the accuracy figures understate the potential performance of set AFW65 for authorship attribution. If the authorship of a book is disputed, and the 'decision' of the experiment was made on the basis of the author to whom most chunks were attributed, then, every experiment reported below would yield the correct result (this was not the case in preliminary experiments with a full set of 104 words, or with a prior set which removed the most common words). In interpreting these results, higher accuracy therefore tends to indicate better reliability – i.e. the degree to which we might expect a correct attribution when only a relatively small number of words are available in the disputed text.

Table V: Summary results of experiments on the five test cases using function word set AFW65 (Table IV).

Test case	**1,000 word chunks**		**2,000 word**		**3000 word**	
	AFW65	**AFW54**	**AFW65**	**AFW54**	**AFW65**	**AFW54**
A	76.14%	87.61	87.74%	**93.82%**	73.7333	88.5822
B	90.12%	90.49	89.50%	86.27%	88.7663	88.4556
C	80.19%	78.83	82.13%	**82.67%**	87.1512	89.0204
D	86.44%	86.98	84.57%	85.21%	82.46405	79.86025
E	89.83%	84.60	88.38%	**90.21%**	87.8107	86.8617

To further refine the set of Arabic function words, we examined the occurrences of each of the 65 words on AFW65 in the dataset, and considered the variance of their frequencies across the set of 2,000 word chunks from different authors. That is, if a function word has a low variance across chunks for different authors, then different

authors tend to use that word with the same frequency, and it may not contribute materially to authorship attribution studies. We found 11 such words with relatively low variance, and composed the function word set AFW54 by eliminating these words. AFW54 comprises the set shown in Table IV, with the following removed: 16, 18, 30, 39, 40, 45, 48, 58, 63, 64, 65. Table V also shows the corresponding results on the five test cases when using the new set AFW54.

As mentioned previously, each trial of each experiment for each test case was able to accurately predict the authorship of each of the test *books*, in the case that we regard the authorship attribution decision as the majority vote of predicted authorship of a book's chunks. In finer detail, the accuracy results (percentage of chunks with accurately predicted authorship) indicate the reliability of the underlying method, which is of particular interest when there is a need to attribute the authorship of a relatively small test body of text. It is not straightforward to compare these accuracy figures with other authorship attribution studies, but we report that they compare very favourably with accuracies reported, for example in [18] for Greek texts using a variety of methods, and in [19] for Dutch texts using a linear discriminate classifier. Finally, given the number of trial runs and case studies, we cannot indicate any statistically significant difference between function word sets AFW65 and AFW54, however each is statistically superior to the original complete set of 104 words.

6. Summary, Discussion and Conclusion

We introduced the use of Arabic function words for use in Arabic authorship attribution and related studies for Arabic texts. Our starting point for a set of Arabic function words was based on collection of 104 common function words reflecting the semantics of the English function words from Mosteller and Wallace [1]. Following a collection of experiments and analyses, using a dataset of Arabic novels, we have refined this to two sets of words AFW65 (Table IV), and AFW54 (Table IV, with 11 words omitted as detailed in section 6). Each of AFW65 and AFW54 was used as the basis to transform a number of Arabic texts into frequency vectors, and the 'performance' of these word sets was assessed by experiments that used a hybrid of an EA and LDA to produce a classifier, and then tested that classifier on unseen data. The resulting performance was clearly in line with results that have been noted for authorship attribution studies in other languages. Set AFW54 is arguably a better choice, however we cannot make that claim with any statistical significance. For the cases considered here, only limited

investigation is reported for assessing the appropriate 'chunk' size. For real applications this will likely depend on several factors, but we have determined (partly from preliminary experiments with smaller chunk sizes) that at least around 1,000-word chunks are necessary to obtain adequate characterization of function word usage for Arabic authors.

Arguably, this work has confirmed that the concept of function words translates suitably well into the Arabic language. In other words, different authors, by and large, use this set of words in sufficiently different ways, enabling us to capture the stylistic fingerprints of individual authors and use these to distinguish between authors.

References

[1] Mosteller, F. and Wallace, D.L., *Inference and Disputed Authorship: The Federalist*, Reading:Addison-Wesley, 1964.

[2] José Binongo., Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution, *Chance*, **16**(2):9—17, 2003.

[3] Patrick Juola, John Sofko, and Patrick Brennan. (2006). A Prototype for Authorship Attribution Studies. *Literary and Linguistic Computing* **21**:169-178

[4] Shlomo Argamon, Shlomo Levitan (2005). Measuring the Usefulness of Function Words for Authorship Attribution. Association for Literary and Linguistic Computing/ Association Computer Humanities, University Of Victoria, Canada.

[5] Kjell, B. (1994). Authorship Determination Using Letter-pair Frequency Features with Neural Network Classifiers. *Literary and Linguistic Computing, 9*, 119 – 124

[6] G. Fung and M. Olvi (2003) The Disputed Federalist Papers: SVM Feature Selection via Concave Minimization (2003).

[7] Brainerd, B. (1975). Statistical anlaysis of Lexical data using Chi-squared and related distributions. *Computers and the Humanities, 9*, 161 – 178

[8] Burrows, J. F. (1987). Word Patterns and Story Shapes: The Statistical Analysis of Narrative Style. *Literary and Linguistic Computing, 2*, 61 -67.

[9] Doležel, L., & Bailey, R. W. (Eds.). (1969). *Statistics and Style*. New York: Elsevier.

[10] D. I. Holmes (1998) The Evolution of Stylometry in Humanities Scholarship, Literary and Linguistic Computing, **13**(3):111-117

[11] D. I. Holmes and R. S. Forsyth (1995) The *Federalist* Revisited: New Directions in Authorship Attribution, *Literary and Linguistic Computing*, **10**(2):111-127

[12] Jiexun Li, Rong Zheng and Hsinchun Chen (2006) From fingerprint to writeprint, *Communications of the ACM*, **49**(4):76—82.

[13] S.G. Efimovich and S.O. Gennadyevich (2003) Automatic search of indicators of text authorship, *Proceedings of the Korea-Russia International Symposium – KORUS 2003*: 185—188.

[14] Mendenhall, T.C. (1887) The Characteristic Curves of Composition, *Science*, IX, 237-249.

[15] Mendenhall, T.C. (1901) A mechanical solution to a literary problem, *Pop. Sci. Monthly*, **60**: 97—105.

[16] Williams, C.B. (1975) Mendenhall's studies of word-length distribution in the works of Shakespeare and Bacon, *Biometrika*, **62**(1): 207—212.
[17] K. Shaker, D. Corne, R. Everson (2007) Investigating hybrids of evolutionary search and linear discriminant analysis for authorship attribution, Proc. 2007 IEEE Congress on Evolutionary Computation, pp. 2071—2077.
[18] E. Stamatatos, N. Fakotakis, G. Kokkinakis (2001) Computer-based authorship attribution without lexical measures, *Computers and the Humanities*, **35**, 193—214.
[19] H. Baayen, H.van Halteren, A. Neiht, F. Tweedie (2002) An experiment in authorship attribution, in Proc. 6th Int'l Conf. on the Statistical Analysis of Textual Data, (JADT).

Bibliography

Bibliography:

[1] D. Khmelev and F. Tweedie, Using Markov Chains For Identification of writers, Literary and Linguistic Computing, 16(3):299-307 (2001)

[2] C. Sanderson and S. Guenter, On Authorship Attribution via Markov Chains and Sequence Kernels, 18th International Conference on Pattern Recognition (ICPR'06) (2006)

[3] G. Fung, The Disputed Federalist Papers: SVM Feature Selection via Concave Minimization, TAPIA '03 conference on Diversity in computing (2003)

[4] F. Peng, D. Schuurmans, V. Keselj and Swan g, Language Independent Authorship Attribution using Character Level Language Models, EACL '03 tenth conference on European chapter of the Association for Computational Linguistics (2003)

[5] R. María Coyotl-Morales, L. Villaseñor-Pineda, M. Montes-y-Gómez and P. Rosso, Authorship Attribution using Word Sequences, CIARP 2006, LNCS4225 855-853 (2006)

[6] E. Stamatatos, N. Fakotakis and G. Kokkinakis, Automatic Authorship Attribution, EACL '99 Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics (1999)

[7] H. Baayen, Hans van Halteren, A. Neijt and F. Tweedie, An experiment in authorship attribution, JADT 2002 6es Journ´ees internationales d'Analyse statistique des Donn´ees Textuelles (2002)

[8] D. Holmes, L. Gorgon and C. Wilson, A widow and her soldier: Stylometry and the civil war. Literary and Linguistic Computing 16 (4): 403-420. (2001)

[9] M. Oakes, Ant Colony Optimisation for stylometry: Federalist Papers, (2004)

[10] J. Nilo, Who Wrote the 15th Book of Oz? An Application of Multivariate Analysis to Authorship Attribution, *Chance*, **16**(2):9—17, (2003)

[11] D. Foster, Author Unknown: on the trail of Anonymous, Macmillan (2001)

[12] M.B. Malyutov, Authorship Attribution of Texts: a review, Electronic Notes in Discrete Mathematics (21) 353–357 (2005)

[13] E. Stamatatos, A survey of modern Authorship Attribution Methods, Journal of the American Society for Information Science and Technology 60(3)538–556, (2009)

[14] D. Holmes, the Evolution of stylometry in humanities scholarships, Literary and Linguistic Computing, **13**(3):111-117 (1998)

[15] F. Mosteller, D. Wallace, Inference and Disputed Authorship: the Federalist Reading:Addison-Wesley,(1964)

[16] F. Tweedie, D. Holmes, T. Corns, The provenance of De Doctrina Christiana Attributed to John Milton: A Statistical Investigation, Literary and Linguistic Computing 13 (2): 77-87 (1998)

[17] D. Estival, Author Attribution with e-mail messages *Journal of Science*, Vietnam National University. pp.1-9. (2008)

[18] T. Reicher, I Kristo, Igor Belsa, A. Silic, Automatic Authorship Attribution for text in Croatian Language Using Combination of Features, KES'10 Proceedings of the 14th international conference on Knowledge-based and intelligent information and engineering systems (2010)

[19] www.BBC.co.uk/Languages/guide.

[20] www.SIL.org

[21] en.wikipedia.org/wiki/Language_family

[22] esl.fis.edu/grammar/langdiff/index.htm

[23] how-to-learn-any-language.com/e/languages/similarities/index.html

[24] en.wikipedia.org/wiki/English_language

[25] en.wikipedia.org/wiki/Arabic_language

[26] en.wikipedia.org/wiki/Afro-Asiatic_languages

[27] K. Shaker, D. Corne, R. Everson, Investigating Hybrids of Evolutionary Search and Linear Discriminant Analysis for Authorship Attribution, in Proc. of IEEE Congress on Evolutionary Computation,(2007)

[28] I. Nabney, Netlab Algorithms for Pattern Recognition

[29] R. A Bosch, and J. A. Smith, Separating hyperplanes and the authorship of the disputed federalist papers. American Mathematical Monthly 105, 7 , 601-608 (1998).

[30] R. M. Everson and J. E. Fieldsend, Multi-Objective Optimisation of Safety Related Systems: An Application to Short Term Conflict Alert (2004)

[31] H.Halteren, F. Tweedie and H. Baayen Outside the cave of shadows: Using syntactic annotation to enhance authorship attribution (1996)

[32] A. Shlomo, L Shlomo.. Measuring the Usefulness of Function Words for Authorship Attribution. Association for Literary and Linguistic Computing/ Association Computer Humanities, University Of Victoria, Canada. (2005)

[33] C. Chung, J. Pennebaker, The Psychological Functions of Function Words, New York: psychological press, social communications :343-359 (2007)

[34] P. Juola, J. Sofko, and P. Brennan. A Prototype for Authorship Attribution Studies. *Literary and Linguistic Computing* **21**:169-178 (2006).

[35] B. Kjell, Authorship Determination Using Letter-pair Frequency Features with Neural Network Classifiers. *Literary and Linguistic Computing, 9*, 119 – 124 (1994).

[36] B Brainerd,. Statistical anlaysis of Lexical data using Chi-squared and related distributions. *Computers and the Humanities, 9*, 161 – 178 (1975).

[37] J. F. Burrows, (1987). Word Patterns and Story Shapes: The Statistical Analysis of Narrative Style. *Literary and Linguistic Computing, 2*, 61 -67.

[38] Doležel, L., & Bailey, R. W. (Eds.). (1969). *Statistics and Style*. New York: Elsevier

[39] D. Holmes and R. Forsyth The *Federalist* Revisited: New Directions in Authorship Attribution, *Literary and Linguistic Computing*, **10**(2):111-127 (1995)

[40] J. Li, R. Zheng and H. Chen From fingerprint to writeprint, *Communications of the ACM*, **49**(4):76—82. (2006)

[41] T. Mendenhall, The Characteristic Curves of Composition, *Science*, IX, 237-249. (1887)

[42] T Mendenhall. A mechanical solution to a literary problem, *Pop. Sci. Monthly*, **60**: 97—105. (1901)

[43] C Williams, (1975) Mendenhall's studies of word-length distribution in the works of Shakespeare and Bacon, *Biometrika*, **62**(1): 207—212.

[44] G. Yule, (1939) On sentence-length as a statistical characteristic of style in prose, with applications to two cases of disputed authorship, *Biometrika* **30**:363—390.

[45]http://www.thenational.ae/thenationalconversation/industry-insights/technology/arabic-potential-left-unread

[46] J. Knowles, D. Corne, K Deb, Multiobjective Problem Solving from Nature, Springer (2008)

[47] C. Bishop, Neural Networks for Pattern Recogenetion, Oxford University press, (1995)